Advant® OCS

Advant Workplace Engineering Methods

Version 1.0

Reference Manual

3BSE 017 088R101

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products AB. ABB Automation Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products AB be liable for direct, indirect, special, incidental, or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

TRADEMARKS

Advant is a registered trademark and AdvaSoft is a trademark of ABB Asea Brown Boveri Ltd., Switzerland.

Microsoft, MS-DOS, Visual Basic and Windows NT are registered trademarks of Microsoft Corporation.

Ghost is a registered trademark of Symantec Corporation.

Instrumentation ActiveX Library is a trademark of Global Magic Software Inc.

Symbol Factory is a trademark of Reichard Software Corporation.

Copyright © ABB Automation Products AB 2000

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 About this Book 1
1.2 Product Overview 2
1.3 Prerequisites and Requirements 3
1.4 Related Documentation 4
1.5 Conventions 6
1.6 Terminology 7
  1.6.1 General 7
  1.6.2 Layout of the Advant Plant Explorer Window 10
  1.6.3 Layout of the Advant Workplace Window 10

Chapter 2 - The ABB Object concept

2.1 ABB Objects 13
2.2 Aspects of ABB Objects 14
2.3 Views on Aspect Data of ABB Objects 15
2.4 Relations between ABB Objects 17
  2.4.1 Functional Relations between ABB Objects 18
  2.4.2 Locational Relations between ABB Objects 19
  2.4.3 Relations between ABB Objects of the Control System 21
  2.4.4 Object Type Structure 22
2.5 Structures for Administration of Advant Workplace Products 23
2.6 ABB Object Type Concept 23

Chapter 3 - The Milko Chemical Example Project

3.1 Overview of the Example Project 27

Chapter 4 - Overview of the Engineering Tasks

4.1 Introduction 31
4.2 Analysis 31
  4.2.1 Domain analysis 32
  4.2.2 Identifying ABB Objects 32

CONTENTS (continued)

4.2.3 Identifying ABB Object Types 32
4.2.4 Identifying structures 32
4.2.5 Identifying Typical solutions 32
4.3 The Engineering Tasks 33
4.4 Creating the Building Blocks 33
4.5 Creating the Structures and an Overview 34
4.6 Creating the Controller Applications 35
4.7 Creating the Remaining Applications 36

Chapter 5 - Typical Solutions Overview

5.1 Introduction 37
5.2 Preparation 38
5.3 Our Example 38
5.4 Name References in a Typical Solution 40
5.5 Building the Drain system typical solution 41
5.5.1 Build the Object Types 42
5.5.2 Build the Functional Structure for the Solution 42
5.5.3 Create the Relative Name Aspects 43
5.5.4 Create the Graphical Elements 44
5.5.5 Moving graphical aspects to the object types 48
5.5.6 Adding Additional Aspects to the Typical Solution 51
5.6 Using the Typical Solution 52
5.7 Limitations in AdvaSoft 2.0 for Windows NT 54

Chapter 6 - Build the Object Types

6.1 Introduction 55
6.2 Creating an Icon Aspect 55
6.3 Creating other aspects 57
6.4 Configuring the Object Type Definition 57
6.4.1 Privilege and Structure Information 58
6.4.2 Inheritance 59
6.4.3 Proposed and Allowed Aspect Categories 60

CONTENTS (continued)

6.4.4 Aspects that are Automatically Created 61
6.4.5 Proposed and Allowed Child Objects 62
6.5 Object types provided with the release 64
6.5.1 3-rd party OPC server support 64
6.5.2 ABB System 65
6.5.3 Alarms and Events 65
6.5.4 Control System 65
6.5.5 Graphics 65
6.5.6 Location 65
6.5.7 Plant & Mill 65
6.5.8 Plant Advant System Specific 65
6.5.9 Project Advant System specific 65

Chapter 7 - Export/Import of Existing Solutions

7.1 Export/Import function 67
7.2 The Milko Chemical example 67
7.3 Exporting the Drain system Solution 68
7.4 Importing Existing Solutions 69

Chapter 8 - Build the Functional Structure

8.1 Creation of ABB Objects in the Functional Structure 71
8.2 Assigning Aspects (application data) to ABB Objects 74
8.2.1 Selecting an ABB Object 75
8.2.2 Creating New Aspects 76
8.2.3 Copying an Aspect 78
8.2.4 Editing an Aspect 78
8.3 Adding Aspects According to Our Example 78
8.4 Process Object Types to be Used in the Functional Structure 79
8.4.1 Sites and Areas, Process Cells, Process Units 80
8.4.2 Equipment modules 81
8.4.3 Control Modules 81

CONTENTS (continued)

Chapter 9 - Build the Location Structure

9.1 Building up the Overview Location Structure 83

Chapter 10 - Build Graphic Displays

10.1 The Milko Milkshake Plant 85
10.2 Checking for Needed Building Blocks 86
10.3 Building the Milko Overview Graphic Display 86
10.4 Building the First Parts of a Milk Reception Display 89
10.5 Building the Rest of the Displays 91

Chapter 11 - Set up a Controller Configuration Project

11.1 Application Builder 93
11.2 Set up a Project in APB 93
11.3 Create the Nodes in APB 96

Chapter 12 - Build the User Application Program

12.1 Overview 99
12.2 Starting the Function Chart Builder 99
12.3 PC Program 101
  12.3.1 Create PC program and the Mandatory PC Elements 102
12.4 Configuration by DB Elements 104
12.5 More Information on DB Elements 109
  12.5.1 Event Detection 109
  12.5.2 Event Transfer 110
  12.5.3 Configuration of Event Sets 111
  12.5.4 Configuration of Extended DB Elements 112
    12.5.4.1 Overview 112
    12.5.4.2 Extended DB Element Communication Between Stations 113
    12.5.4.3 Event Handling with Extended DB Elements 115

CONTENTS (continued)

Chapter 13 - Generate and Download Target Code

13.1 Generate Target Code 117
13.2 Backup and Restore 118
13.3 Target Control Functions 120

Chapter 14 - Restart the OPC Server

14.1 General 123
14.2 Restart the OPC Server 124

Chapter 15 - Build the Control Structure

15.1 Populating the Control System Structure 125
15.2 Removing Obsolete Objects 128
15.3 Verifying the Result of an Upload 128
15.4 Available AC 100 Series Object Types 130

Chapter 16 - Complement the Functional Structure

16.1 Complementing the Functional Structure with Signal Objects 131
16.2 Recommendations 133

Chapter 17 - Complement the Location Structure

17.1 Complement the Location Structure with Signal Objects 135
17.2 Recommendations 137

Chapter 18 - Building Alarm and Event Applications

18.1 Alarm and Event overview for AF100 139
18.2 Alarm and Event defaults 141
18.3 Creating full Alarm and Event Lists 145
18.4 Creating Alarm Lists for Important Plant Sections 148
18.5 Configuring the Alarm Band 149
18.6 Configuring Alarm Logging 149

CONTENTS (continued)

Chapter 19 - Customizing User Environment, Advant Workplaces

19.1 Customizing the User Environment 151
19.2 Set-up of Direct Navigation to Overview Displays 153
  19.2.1 Display Shortcuts in the Advant Workplace 153
  19.2.2 Configuring Alarm Bands - short-cuts to Alarm and Event lists 154
  19.2.3 Display Shortcuts in the display bar 156
19.3 Building up Your Own Menu Display 159
19.4 Setting up the Final Users 162
  19.4.1 General 162
  19.4.2 Creating NT Users 163
  19.4.3 Creating the Advant Users 163

Chapter 20 - Building History and Trend Applications

20.1 Building History Applications 167
20.2 Building Trend Applications 169

Chapter 21 - Building Complementary Applications

21.1 Adding a Web Aspect showing Technical Data for a Valve 172
21.2 Add an ActiveX Wrapper Aspect 174
21.3 Adding a Windows Application Aspect opening a Word document 175

Chapter 22 - Backup of Application Data

Appendix A - Example on Reference Material

A.1 Milko Milkshakes 179
A.2 Overview 179
A.3 Process Cells versus Common Resources 180
A.4 Milkshake Plant Process Descriptions 180
A.5 CIP 185
A.6 S88 Analysis 189
A.7 Object Count 197

CONTENTS (continued)

Appendix B - Advant Structures

B.1 Advant Structures 205
  B.1.1 Admin Structure 206
  B.1.2 Aspect System Structure 207
  B.1.3 Control Structure 209
  B.1.4 Functional Structure 210
  B.1.5 Graphics Structure 212
  B.1.6 Library Structure 213
  B.1.7 Location Structure 213
  B.1.8 Node Administration Structure 214
  B.1.9 Object Type Structure 215
  B.1.10 Service Structure 216
  B.1.11 User Structure 217
  B.1.12 Workplace Structure 218

INDEX

Chapter 1 Introduction

1.1 About this Book

This manual is intended as a guide for plant engineering using Advant Workplace Products software together with the Advant controller 100 Series. Instructions for operating the various Advant controller engineering tools are provided in other manuals, described in Section 1.4, Related Documentation.

For a description of all installation consult *Advant Workplace Products Administrator´s Guide*.

The book is organized into the following chapters:

- Chapter 1, Introduction
- Chapter 2, The ABB Object concept
- Chapter 3, The Milko Chemical Example Project
- Chapter 4, Overview of the Engineering Tasks
- The rest of the book is structured into a number of tasks. See Chapter 4, Overview of the Engineering Tasks for an overview of these tasks.
  See Chapter 5 to 22 for a description of the various tasks.

Use of Warning, Caution, Information, and Tip

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

⚠ Warning indicates the presence of a hazard which could result in *personal injury*.

❗ Caution indicates the presence of a hazard which could result in *equipment or property damage*.

Information alerts the reader to pertinent facts and conditions.

Tip indicates advice on, for example, how to design your project or how to use a certain function.

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

1.2 Product Overview

The Advant Workplace Products is a family of products, for process monitoring and control. This release consists of the following main product:

- **AdvaSoft 2.0 for Windows NT**
  Contains the functionality your operators need for efficient control and supervision of your process. Key functionality is presentation of graphic displays, execution of process dialogs, presentation of trends, and alarms.

Options:

- **Advant Graphic Builder**
  Contains the functionality your engineers need for efficient and cost effective engineering of the graphic displays.
- **Advant History**
  Contains the functionality you need to Historize the process object attributes of your plant. Key features are long term storage of process values (utilizing compression techniques) and data refinement. The data can be presented, for example as trends in a graphic.

- **Advant Connectivity Library for Advant controller 100 Series**
  This option provides ready made graphic elements and process dialogs for Advant Fieldbus 100. It includes the ABB process knowledge in the form of Objects types and process control aspects. Use them for an efficient engineering of your displays and dialogs.
- **Advant Graphic Library for Process Industry**
  This library provides you with a number of general graphic elements suitable for use together with Advant Workplace Products and Advant controller 100 Series objects.
- **Symbol Factory**
  The Symbol Factory consist of more than 3000 symbols for the process industry.
- **Instrumentation ActiveX Library**
  This library provides you with 11 powerful ActiveX Controls for automation.

The following products are required for connection to Advant Fieldbus 100:

- **Advant OPC Server for Advant Fieldbus 100**
  Gives you connection to the Advant controller 100 series controllers via the Advant Fieldbus 100 communication interface.
- **Advant Fieldbus 100 hardware**

  You can order the necessary hardware in 4 different packages depending on your choice of cable type and redundancy.
- **AMPL Control Configuration**
  AMPL Control Configuration provides the engineering tools for Advant controller 100 Series.

1.3 Prerequisites and Requirements

For a description of the system requirements consult the *Advant Workplace Products Administrator´s Guide*.

To perform the tutorials you normally need to be logged in as a user with **Application Engineer** privileges.

1.4 Related Documentation

Table 1-1 lists documentation related to the content of this reference manual.

Table 1-1. Related Documentation

Category	Title	Description
System Administration	Advant Workplace Products Administrator's Guide	This book describes how you install the Advant Workplace Products software, including software features, and how you then perform maintenance.
Software	Advant Workplace User's Guide	This book describes everything about the user interface, for example how you present information, navigate and control objects.
	Advant Plant Explorer User's Guide	This book describes how you build hierarchically structured models of a plant or project as well as how you navigate in the structures.
	Advant Graphics User's Guide	This book describes how you build and use graphics in Advant Workplace Products. It includes information about the basic building blocks provided by ABB.
	Advant Alarms User's Guide	This book explains how to use the Advant Alarm and Event functions. It covers alarm and event configuration and presentation.
	Advant Trend User's Guide	This book describes how to use the trend graphs to view History data during runtime.
	Advant Reports Add-Ins User's Guide	This book describes how to include historical information in process reports

Table 1-1. Related Documentation (Continued)

Category	Title	Description
Options	Advant Connectivity Library for Advant Controller 100 Series Reference Manual	This book is a set of pre-defined graphic elements for Advant Controller 100 Series.
	Advant Graphic Library for Process Industry Reference Manual	This book is a set of pre-defined graphics used by the process industry.
	Advant OPC Server for Advant Fieldbus 100 User’s Guide	This guide describes the Advant OPC Server for Advant Fieldbus 100 (Advant OPC Server for AF100), which is a standard OPC interface for Advant Fieldbus 100.
	Advant History User’s Guide	This book describes the Advant History software.
Other	Advant Workplace Products for Operators Product Guide	This book is a presentation of the software and software options in Advant Workplace Products for Operators. It also includes hardware and software requirements and specifications.
	Application Builder User’s Guide	Describes the installation of functions and components for control configuration and contains the operating instructions of Application Builder, Bus Configuration Builder and support functions.
	Function Chart Builder User’s Guide	Contains a description of AMPL application programs configuration.
	AdvaSoft 2.0 for Windows NT Operator’s Guide	This is the operators manual and contains information on how to use AdvaSoft 2.0 for Windows NT.

1.5 Conventions

The following conventions are used throughout this book for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
- A plus sign is used to indicate that you must simultaneously press several keys. For example, CTRL+C indicates that you must hold down the CTRL key while pressing the C key.
- The phrase “press and release” is used to indicate that you sequentially press several keys. For example, to close a window, press and release ALT, and then F4. This indicates that you press and release the ALT key, and then you press and release the F4 function key.
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
    - The following convention is used for menu operations: MenuName > MenuItem > CascadedMenuItem. For example: choose **File** > **New** > **Type**.
    - The **Start** menu name always refers to the **Start** menu on the Windows NT Task Bar.
- System prompts/messages in the MS-DOS Prompt are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

  ```
  Entered value is not valid. The value must be 0-30.
  ```

  In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

  ```
  TIC132
  ```

1.6 Terminology

1.6.1 General

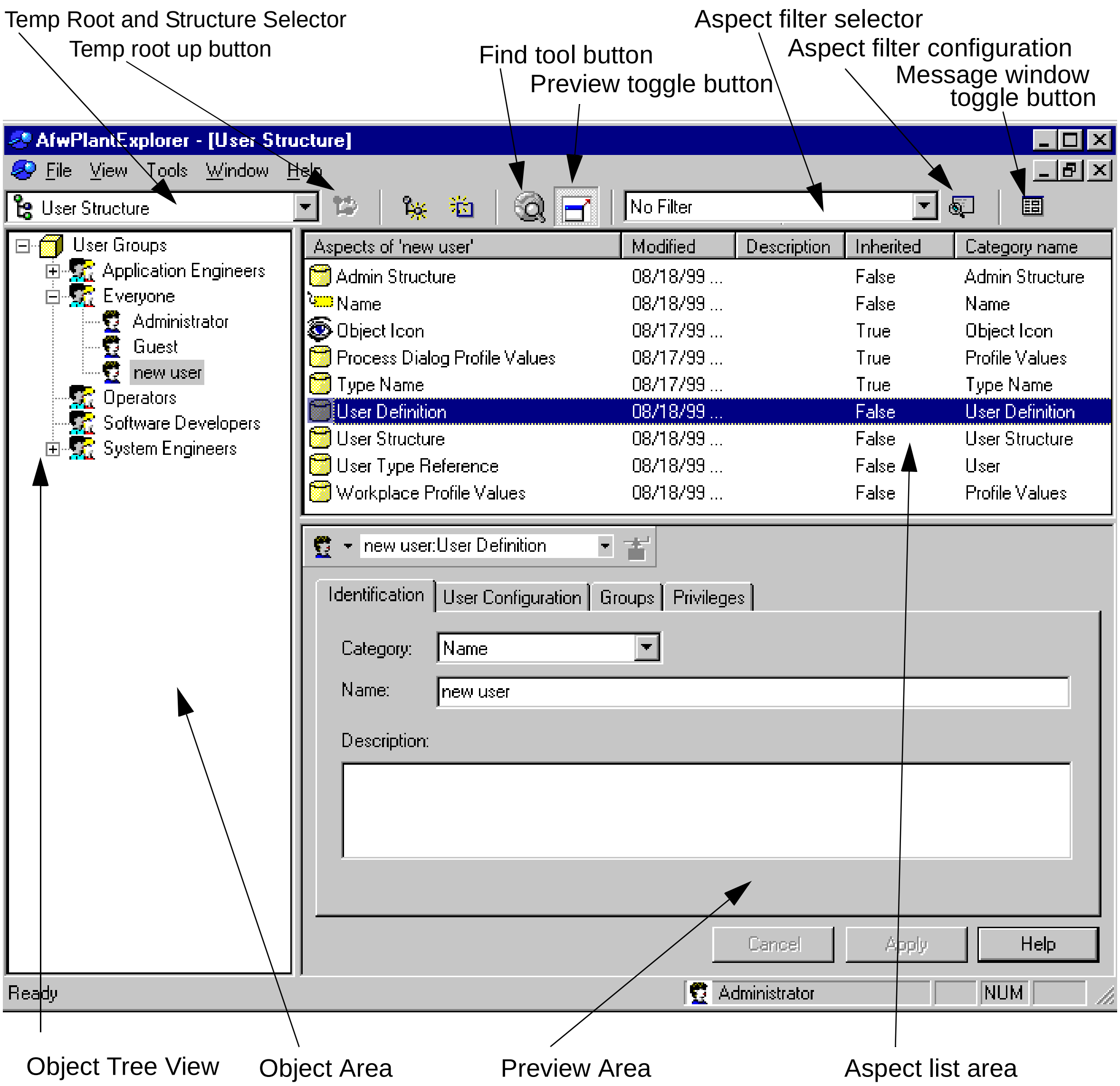
The following is a list of terms that you should be familiar with.

Term	Description
Advant controller 100 Series	The Advant controllers AC70, AC100 and AC160. (AC80 is only partly supported.)
ABB Object	A representation of a real object that can be seen and operated in the Advant System. ABB Objects are organized into *structures*. An ABB Object is described by its *aspects*. Examples of ABB Objects: plant equipment, components and devices, location entities, information entities.
Advant Plant Explorer	Advant Plant Explorer is a tool for creating the building blocks that you use to put together the parts of a plant or a project model. You can also use it to browse the structures of a plant or project, and to find particular objects contained in the structures by means of the **Find** tool.
Advant user	An Advant user, for example an operator or application engineer, performs different Advant Workplace operations. Advant users can be placed into one or more Advant user groups. A Windows NT user, which is used when logging into a node (automatically or explicitly), can automatically be promoted to an Advant user if the system is so configured. The user and user group concept is used to handle personal customizing, authority control and operation logging of what the users have been doing in the system.

Term	**Description**
Advant user group	An Advant user group is a set of Advant users, for example operators or application engineers. All members within an Advant user group have the same privileges. For example, everyone belonging to an Advant user group called Operator, is allowed to operate all pieces of equipment within a plant. Notice that the Advant user can be a member of several Advant user groups at the same time. NOTE! The Advant user and Advant user group concept matches the user and group concept found in Windows NT but some features have been added for process specific purposes.
Advant User Privileges	An Advant User holds a privilege if any of the user groups the user belongs to has the privilege assigned. There are four predefined privileges: Operator, Application Engineer, System Engineer and Software Developer.
aspect	An aspect is a description of an ABB Object (which in itself represents a real object, for example a valve). The ABB Object keeps the aspect data together in order to give the user an easy and consistent access to the information. Examples of aspects: real time video, functional description, circuit diagram, OCS control logic. An aspect has, in most cases, also several views.
Module	A piece of hardware for example an I/O module
Operator mode	One of the two modes of operation. This mode is used for Operator Workplaces. The whole screen is covered by the main Workplace Window. There is no way to resize or minimize/maximize the window. Icons, normally placed on the desktop, are covered by the main Workplace Window (and therefore not visible). Overlapping Workplace Windows are always placed on top.

Term	**Description**
structure	A hierarchical organization of ABB Objects that describes the dependencies between the real objects. Several structures can exist and an ABB Object can exist in more than one structure. You can create several structures depending on your needs. Some predefined structures are available.
Windows™ mode	One of the two modes of operation. In Windows mode all windows act as normal NT-windows. You may change between minimized window, normal window and full-screen mode in the same way as with any Windows window by using the three buttons in the upper right hand corner of the window. You may also resize and move around the window on the screen. This makes it possible for you at all times to resize and place the main Workplace Window according to your needs.

1.6.2 Layout of the Advant Plant Explorer Window



User Groups
- Application Engineers
- Everyone
  - Administrator
  - Guest
  - new user
- Operators
- Software Developers
- System Engineers

Aspects of 'new user'	Modified	Description	Inherited	Category name
Admin Structure	08/18/99 ...		False	Admin Structure
Name	08/18/99 ...		False	Name
Object Icon	08/17/99 ...		True	Object Icon
Process Dialog Profile Values	08/17/99 ...		True	Profile Values
Type Name	08/17/99 ...		True	Type Name
User Definition	08/18/99 ...		False	User Definition
User Structure	08/18/99 ...		False	User Structure
User Type Reference	08/18/99 ...		False	User
Workplace Profile Values	08/18/99 ...		False	Profile Values

new user:User Definition

Identification | User Configuration | Groups | Privileges

Category: Name

Name: new user

Description:

Figure 1-1. Stand-alone Advant Plant Explorer window

1.6.3 Layout of the Advant Workplace Window

At start you will get a pre-configured start layout with an Application Bar and a Status Bar.

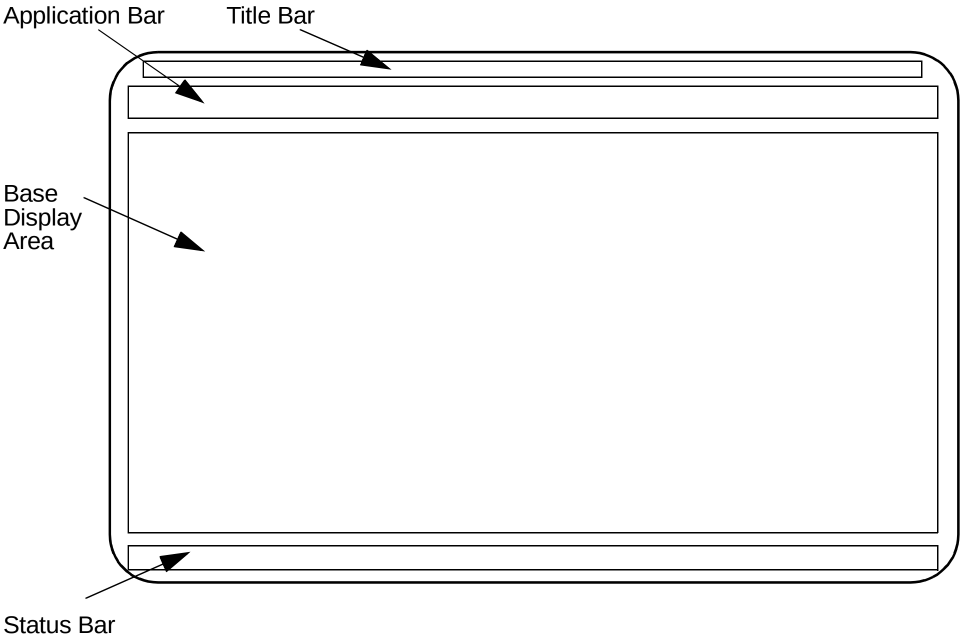


Figure 1-2. Advant Workplace Windows Four Areas

Chapter 2 The ABB Object concept

2.1 ABB Objects

All information handled during the engineering process is assigned to **ABB Objects**.

The ABB Objects handled in the engineering process represent physical objects or virtual objects, for example, functions, and are used as a container for engineering and run time data. An ABB Object is a data model for an entity treated in a design, engineering, operation and maintenance process.

The Advant Workplace Products and the corresponding engineering tools support a concept of plant modeling by using ABB Objects.

ABB Objects can be grouped into three groups: Signal Objects, Process Objects and System Objects:

- **Signal Objects:** The objects corresponding to I/O signals. In AF100: AIS, DIS, DOS, AOS, DAT_AI, DAT_AO, DAT_DAT, MB, MBS, MI, MIL, MR
- **Process Objects:** More complex object types representing
  - The process such as: Blender, Feeder, Mixer, Pump, Reactor, Silo/Tank, Valve, Motor etc.
  - Location such as area, building, room, cabinet etc.
- **System Objects:** For example: Network object, Node object, Workplace object, User object, Service Group object etc.

2.2 Aspects of ABB Objects

The engineering data describing an ABB Object is grouped into aspects. An aspect is an assembly of data that describes certain properties of an ABB Object, such as:

- Functional properties
- Physical construction properties
- Location properties.

An ABB Object may have several aspects, it has at least one aspect which describes the Basic Properties of the ABB Object, for example, name and type. Each aspect is handled by assigned tools.

Figure 2-1 illustrates the definitions ABB Object and aspect.

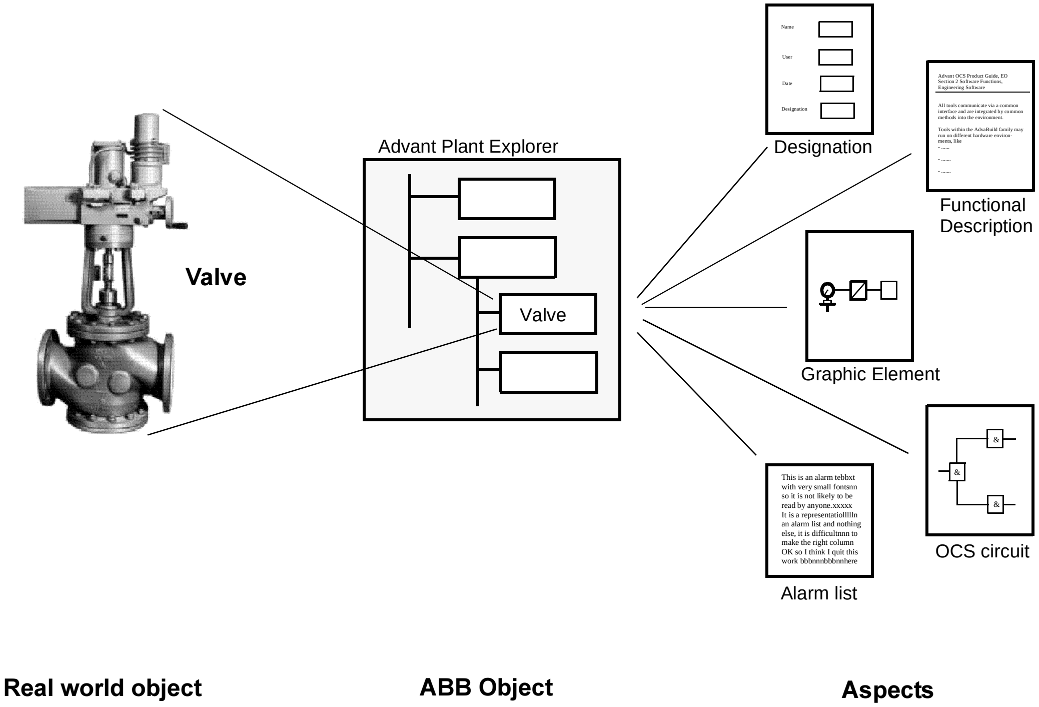

Name
User
Date
Designation
Designation
Advant OCS Product Guide, EO
Section 2 Software Functions,
Engineering Software
All tools communicate via a common interface and are integrated by common methods into the environment.
Tools within the AdvaBuild family may run on different hardware environments, like
- ......
- ....
- ....
Functional Description
Advant Plant Explorer
Valve
Valve
Graphic Element
&
&
&
OCS circuit
This is an alarm tebbxt with very small fontsnn so it is not likely to be read by anyone.xxxxx It is a representatiollllln an alarm list and nothing else, it is difficultnnn to make the right column OK so I think I quit this work bbbnnnbbbnnhere
Alarm list
Real world object
ABB Object
Aspects

Figure 2-1. Illustration of an ABB Object with its aspects

2.3 Views on Aspect Data of ABB Objects

Aspect data are presented in views. An aspect may have several views. Opening aspect views gives access to the relevant aspect system with the relevant tool to inspect, modify and map the data.

The data and their mapping may be presented, for example, as:

- Lists
- Tables
- Diagrams (Circuit Diagram)
- Drawings (Arrangement Drawing)
- Graphics

Views are predefined in the Advant Workplace Products. Each aspect has one or several pre-defined views. Figure 2-2.

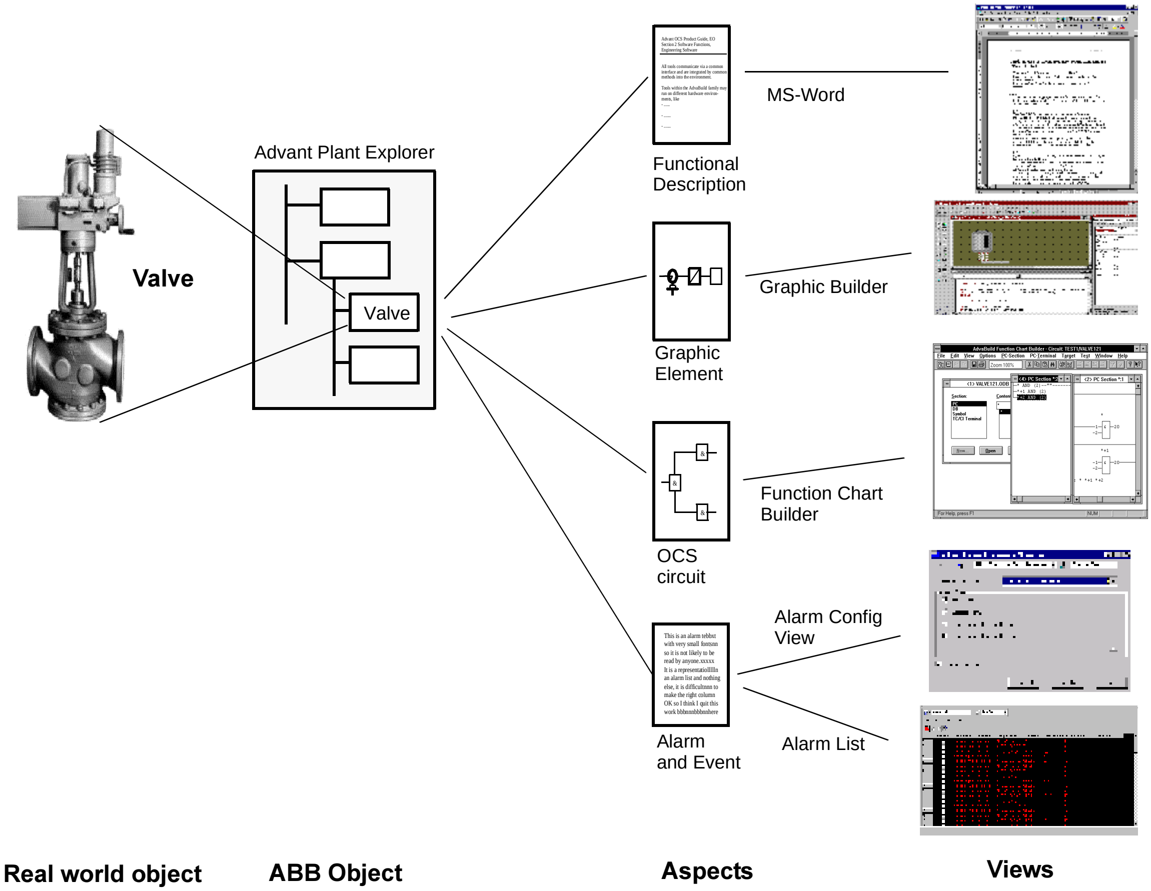


Figure 2-2. Illustration of an ABB Object with its aspects

2.4 Relations between ABB Objects

The relations between ABB Objects are established by placing them in one of the following predefined Engineering structures for ABB Objects:

- Functional structure
- Location structure
- Control system structure
- Object type structure

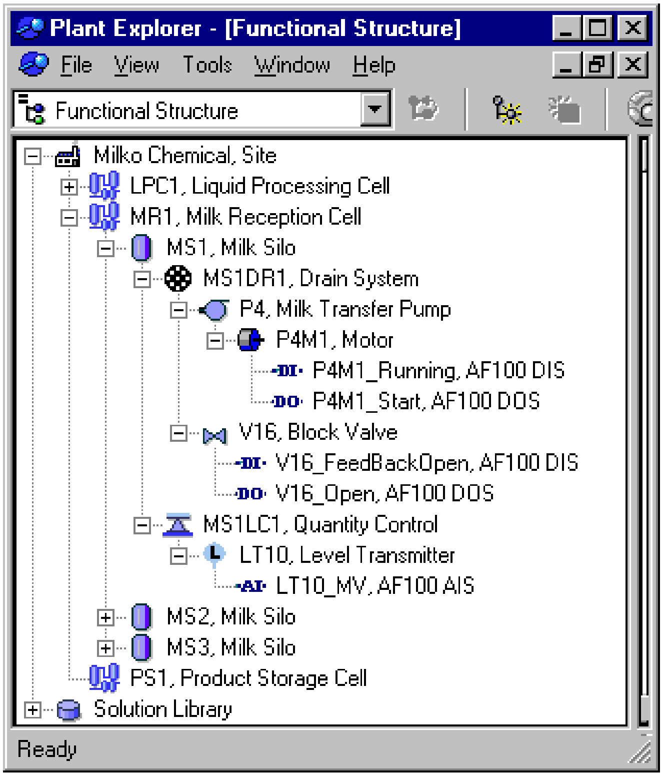
Placing ABB Objects into these predefined structures supports the navigation in the engineering data and the designation of the ABB Objects in the plant and the documents. You may also use the structures as an operator or maintenance person for fast navigation to and location of certain information.

You may also create and use your own structures.

2.4.1 Functional Relations between ABB Objects

Functional relations between ABB Objects are presented in a function oriented structure. This structure is built up by placing ABB Objects into this structure according to their hierarchical relations.

The following example shows a number of ABB Objects representing a Storage Tank with it's drain system and quantity control:



- Milko Chemical, Site
  - LPC1, Liquid Processing Cell
  - MR1, Milk Reception Cell
    - MS1, Milk Silo
      - MS1DR1, Drain System
        - P4, Milk Transfer Pump
          - P4M1, Motor
            - P4M1_Running, AF100 DIS
            - P4M1_Start, AF100 DOS
        - V16, Block Valve
          - V16_FeedBackOpen, AF100 DIS
          - V16_Open, AF100 DOS
      - MS1LC1, Quantity Control
        - LT10, Level Transmitter
          - LT10_MV, AF100 AIS
    - MS2, Milk Silo
    - MS3, Milk Silo
  - PS1, Product Storage Cell
- Solution Library

Figure 2-3. Functional relation of a storage tank

2.4.2 Locational Relations between ABB Objects

Locational relations between ABB Objects are presented in a location oriented structure. Distinguish between topographic locations like:

- Areas
- Buildings
- Rooms

And mounting locations like:

- Cabinets
- Desks
- Racks
- Slots

The topographic location structure and the mounting location structure can be created independently from each other and connected in a later planning stage. The connection is performed by placing mounting location structure trees into the topographic location structure. The mounting location structure contains a number of ABB Object structure trees that represent for example the cabinets, desks and other constructional units of the plant.

For Advant Controller 100 Series the latter is not that essential to present since the Controllers are not very complex. Therefore, in AdvaSoft 2.0 for Windows NT, location structure means topographic location.

Figure 2-4 shows the principles of a location oriented structure.

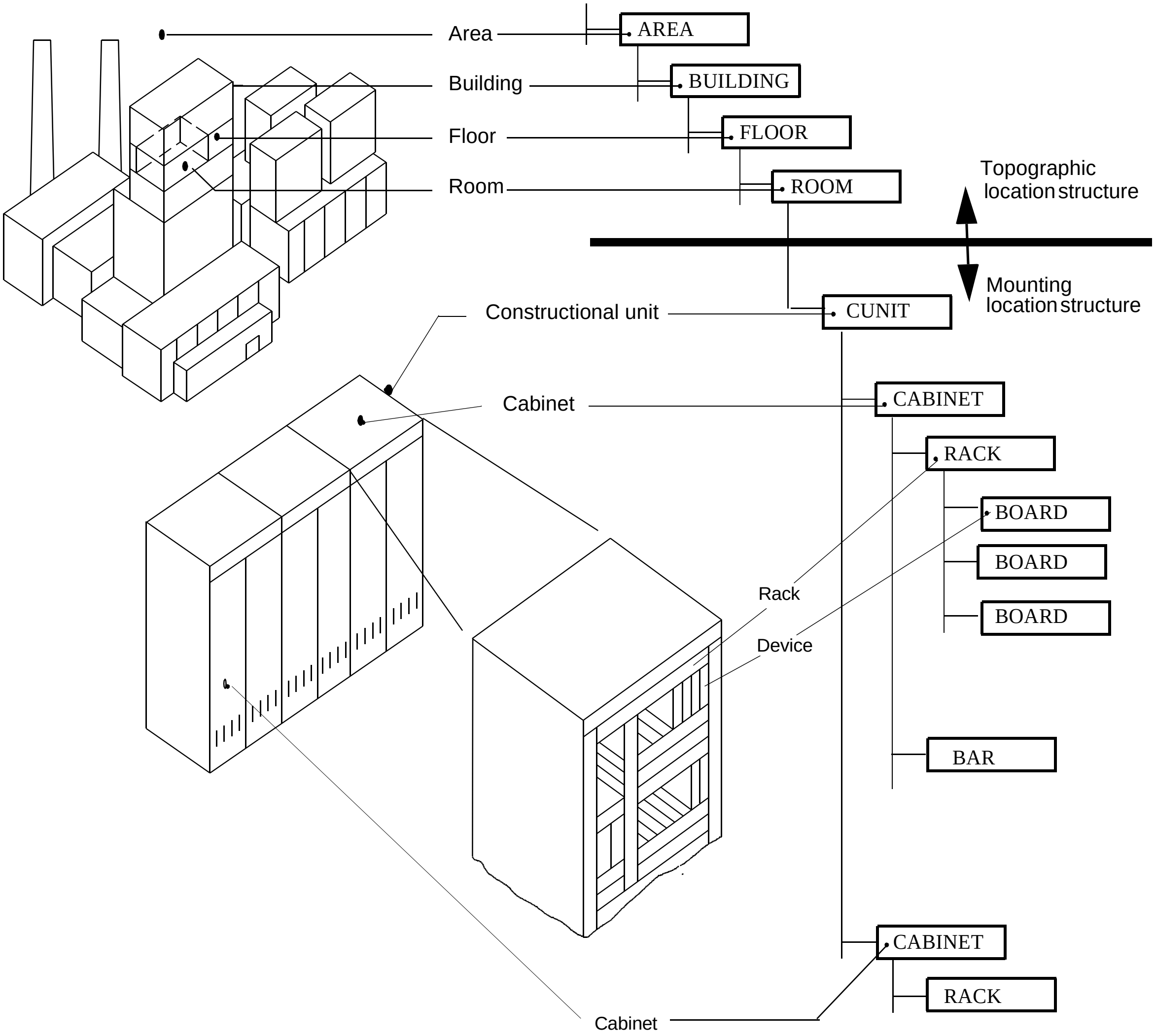


Figure 2-4. Principle of a location oriented structure

2.4.3 Relations between ABB Objects of the Control System

The relations between ABB Objects representing the control system are presented in a Control system structure. The structure is built up by placing the relevant ABB Objects into this structure type.

Figure 2-5 shows the principles of structuring ABB Objects that are constituents of the control system.

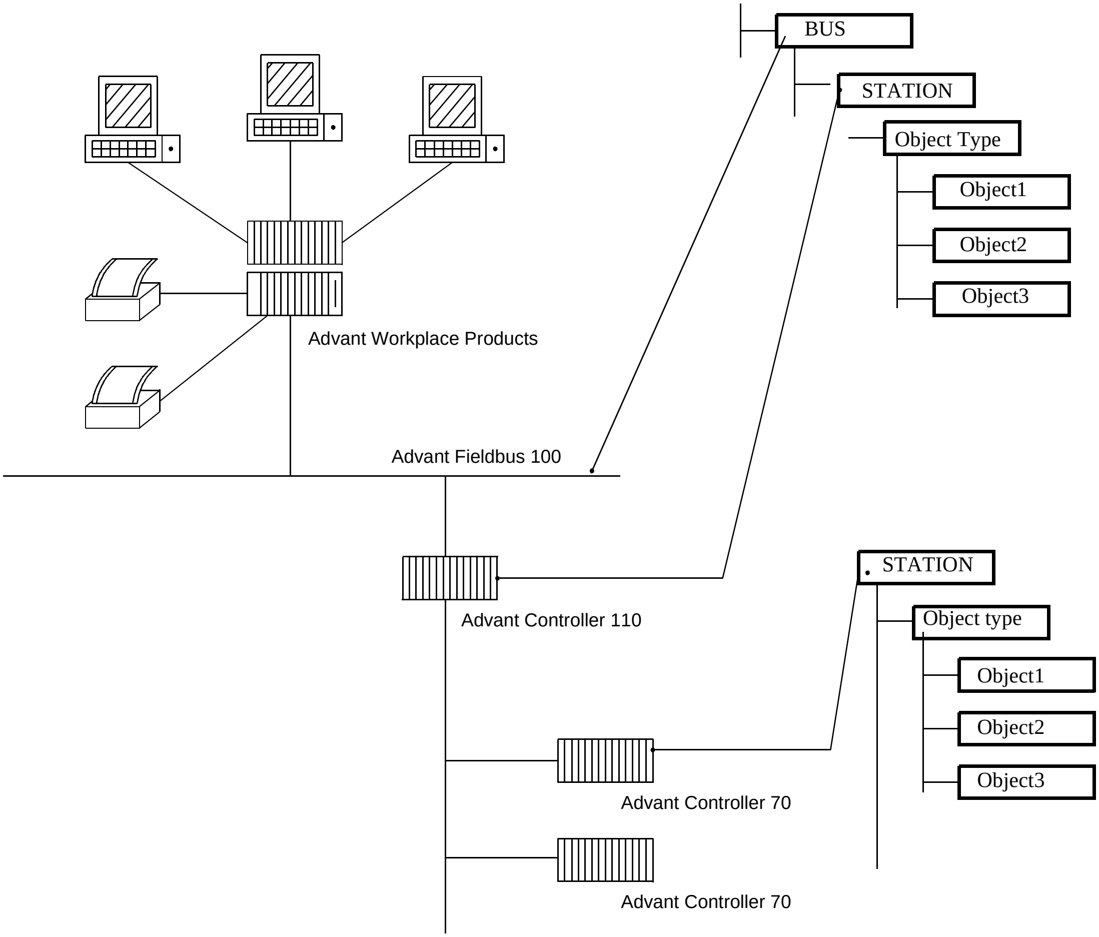
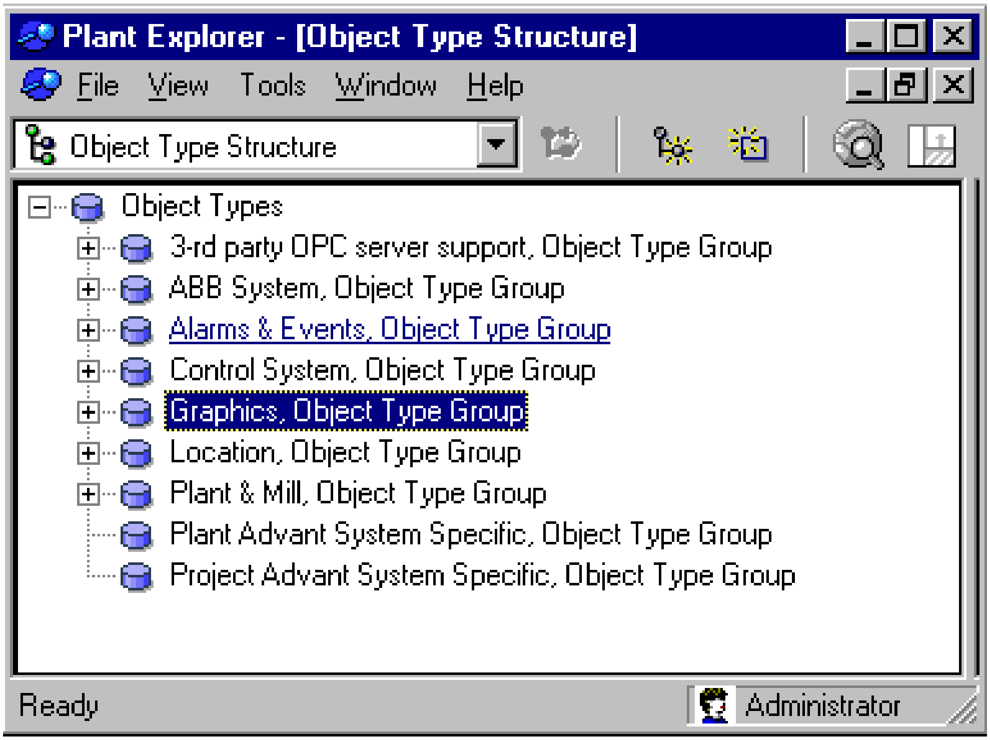


Figure 2-5. Principle of a Control system oriented structure

2.4.4 Object Type Structure

All ABB Objects are instances of an object type. All Object Types are organized in an Object type structure which makes it easier to see dependencies, inheritance etc. See Figure 2-6.

You need to be an application engineer to access this structure.



- Object Types
  - 3-rd party OPC server support, Object Type Group
  - ABB System, Object Type Group
  - Alarms & Events, Object Type Group
  - Control System, Object Type Group
  - Graphics, Object Type Group
  - Location, Object Type Group
  - Plant & Mill, Object Type Group
  - Plant Advant System Specific, Object Type Group
  - Project Advant System Specific, Object Type Group

Figure 2-6. An example of an Object type structure

See Section 6.5, Object types provided with the release for more details about the existing object types.

See also Section 2.6, ABB Object Type Concept for details on the object type concept.

2.5 Structures for Administration of Advant Workplace Products

So far we have described the structures that are used in your engineering and operation of a plant. There are however also other structures, mainly used for the administration and engineering of the Advant Workplace Products. You do not use all of these as an end user but they are all used by the Advant system. As a system administrator you should at least know about for example the Node Administration structure, User structure and Workplace structures.

Appendix B, Advant Structures gives a brief description of all structures in the system from an Advant Workplace Product perspective.

2.6 ABB Object Type Concept

2.6.1 Overview

The ABB Object Type concept provides you with a powerful template function. The Object Type concept defines a number of things, such as:

- Aspects that are automatically created when an Object instance is created.
- Aspects that are suggested as candidates for an Object. (You can describe them as mandatory or not)
- Aspects that are valid for an Object.
- Aspects that are shared by inheritance.
- Rules for how aspects can be modified.
- Children object types that are allowed to be placed under the object in a structure (can be specified for each structure)
- Children object types that are suggested as candidates for you when you create new child objects.

To each of the Object types is attached a **Type Name** aspect.

See Chapter 6, Build the Object Types for examples and details on how to build an object type.

2.6.2 Creating a New Object Instance

As a result of the Object Type concept you have a lot of functionality as soon as you create an ABB Object instance.

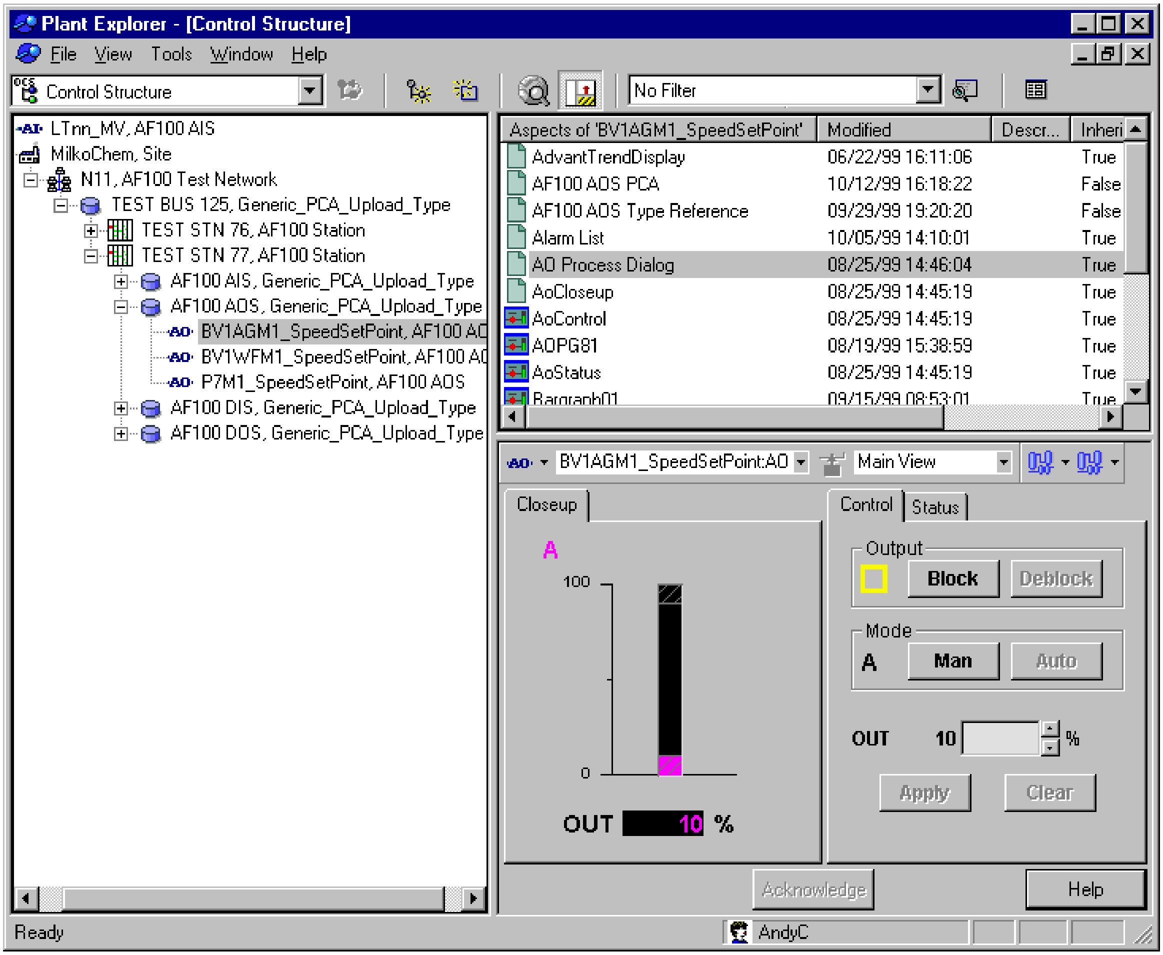
Example: A Valve Process Object. Underneath it in a Functional structure are one DI and one DO signal object. It needs the following default aspects:

- A graphic element aspect
- A process dialog aspect.
- An alarm list aspect - providing an alarm list for all alarms from the connected DI and DO signal objects

Example: Functionality automatically available for a typical signal object through inheritance from its object type is:

- A process control aspect
- A graphic element
- A close up display
- A process dialog aspect
- An alarm and event list aspect - providing a list of alarms for this specific process object only
- A trend aspect - providing an on-line trending of the signal, or if there is a history log aspect on the “Value” attribute, providing trending of the History data together with an on-line trending.

These can be executed directly giving you a view over the object or the ability to control the object.



Control Structure

No Filter

- LTnn_MV, AF100 AIS
- MilkoChem, Site
- N11, AF100 Test Network
  - TEST BUS 125, Generic_PCA_Upload_Type
    - TEST STN 76, AF100 Station
    - TEST STN 77, AF100 Station
      - AF100 AIS, Generic_PCA_Upload_Type
      - AF100 AOS, Generic_PCA_Upload_Type
        - BV1AGM1_SpeedSetPoint, AF100 A
        - BV1WFM1_SpeedSetPoint, AF100 A
        - P7M1_SpeedSetPoint, AF100 AOS
      - AF100 DIS, Generic_PCA_Upload_Type
      - AF100 DOS, Generic_PCA_Upload_Type

Aspects of 'BV1AGM1_SpeedSetPoint'	Modified	Descr...	Inheri
AdvantTrendDisplay	06/22/99 16:11:06		True
AF100 AOS PCA	10/12/99 16:18:22		False
AF100 AOS Type Reference	09/29/99 19:20:20		False
Alarm List	10/05/99 14:10:01		True
AO Process Dialog	08/25/99 14:46:04		True
AoCloseup	08/25/99 14:45:19		True
AoControl	08/25/99 14:45:19		True
AOPG81	08/19/99 15:38:59		True
AoStatus	08/25/99 14:45:19		True
Bargraph01	09/15/99 08:53:01		True

BV1AGM1_SpeedSetPoint:AO

Main View

Mode: A

OUT 10 %

OUT 10 %

Ready

AndyC

Figure 2-7. Example of a Control structure with a selected object

Furthermore there are lists of the most probable aspect categories that you want to add so you get the best possible guidance when you extend your object with more aspects and rules to prevent you from selecting aspect categories that cannot be used with that object.

Finally, when you build your Object structure, you are guided by suggestions and rules for how to build up the children to the object.

2.6.3 Creating New Object Types

One way of using the power of object types is to create typical solutions in the form of new object types. More details are provided in Chapter 6, Build the Object Types.

Chapter 3 The Milko Chemical Example Project

3.1 Overview of the Example Project

A milk products plant, Milko Milkshakes, will be used as an example to explain the engineering process in this manual. The result of this engineering process shall be all the data and documents that are needed for:

- Installation
- Commissioning
- Operation
- Maintenance.

See Figure 3-1 for an overview of the process used in Milko Milkshakes. As you will see in the stepwise description in this document, we have used the ABB Object concept for the representation of the process.

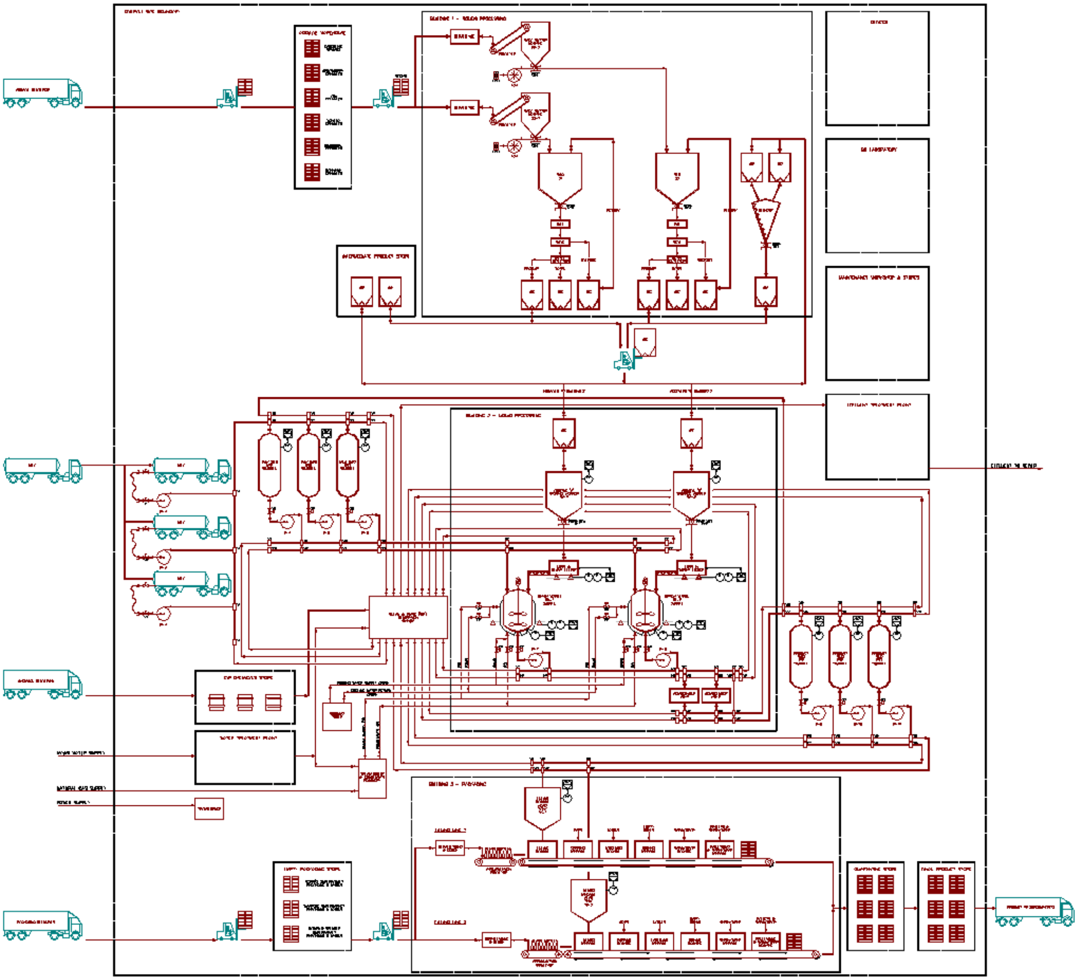


Figure 3-1. Plant overview over Milko Milkshakes - CAD drawing

3.2 Basic Information to Start With

The scope of supply and the necessary engineering task for a project is usually defined in a contract. You find the basic input data on which the engineering is done in documents from the customer (which may be yourself) or project partner. These documents describe the plant, the processes running the plant and specify basically the functions that shall be implemented to run the processes.

The documents that might help you are:

- Process Diagram and Process Description
  (see Appendix A, Example on Reference Material)
- P&I-Diagram
- List of Measuring Points
- Motor and Component List

If you are the customer you need to create these documents yourself or with the help of consultants. If you are not the end customer these documents need to be delivered by the customer or by a partner with a certain status. They may get further revised during the engineering process. These revisions of the documents that specify the basic functions are normally handled by the supplier of the electrical and control equipment in cooperation with the customer.

The import of data that were created externally is managed by import functions.

In our example project we will refer to the activities and to the main steps of the engineering tasks. The focus is put on the tasks for the function oriented and control system oriented engineering. The engineering tasks are described in Chapter 4, Overview of the Engineering Tasks.

Chapter 4 Overview of the Engineering Tasks

4.1 Introduction

The Engineering of the Advant Workplace Products consists of a number of tasks and incremental steps within those tasks. A number of figures in this chapter show how the tasks are described in this book. Before studying them please remember that the order in which you do the tasks does not matter that much. Many tasks can be done in parallel and many can be done in a different order. When your experience grows you will also realize where you can do them in parallel or in another order and where not.

There is one exception to the above statement that the order does not matter that much: The Functional structure should always be built early, preferably first, or as a parallel task to the creation of typical solutions. The sooner you have a complete and correct Functional structure, the better can you get that essential overview that helps you to avoid mistakes. However, as you continue to read this book you will also realize that the functional structure is built using typical solutions and object types built during the project. So, you can start to build the overall structure and then complement the details as the various building blocks are completed.

The ABB Object concept is unique in its ability to structure information for suitable entities. An entity, for example a process cell, should be placed in one branch in the Functional structure and have graphic display aspect, alarm list aspect, event list aspect and so on. These contain the essential information from that branch.

4.2 Analysis

Before you start the engineering work you need to make an analysis of the system that you are going to build. This analysis can be split into five parts:

- Domain analysis
- Identifying ABB Objects
- Identifying ABB Object Types
- Identifying structures
- Identifying Typical Solutions

4.2.1 Domain analysis

The intention of the domain analysis is to identify entity objects and structures in the domain. This analysis should be based on existing standards for how systems are composed into different structures for example IEC61346. See www.iec.ch for details.

Normally you would start with the top object in the functional structure and start to decompose it according to functionality, thus creating a functional structure of objects.

4.2.2 Identifying ABB Objects

The found domain entity objects are then mapped into ABB Objects. The ABB Objects are used as containers for the domain entity objects. their aspects will correspond to the functionality of each respective entity.

4.2.3 Identifying ABB Object Types

If applicable, the found domain entity objects are mapped into existing ABB Object types. You should also identify which object types that need to be developed.

4.2.4 Identifying structures

The ABB Objects are also put into structures. Use for example PowerPoint organization charts or an XML editor.

4.2.5 Identifying Typical solutions

If applicable, functionality that occurs in several parts of the plant should be identified and described as a typical solution based on ABB objects organized in a (functional) structure. These are later on either imported (if they exist already) or developed.

4.3 The Engineering Tasks

For visualization we have grouped the activities in four groups. You don´t nedd to however, to do the tasks done to this. In fact you can do most tasks anytime during the engineering phase and you can also go back to a task and complement it.

A lot of the initial work is done with the engineering tools for the Controller applications. The overview principles for this are described in this book but the details and reference material is found in the User's Guide for the respective functions.

4.4 Creating the Building Blocks

After the analysis you can start to build or import the building blocks. The reasons for you to create building blocks are:

- Each building block that you can identify and that can be used repeatedly will save engineering work.
- You get a methodology with which you can brake the process down into smaller parts, easier to grasp and handle.

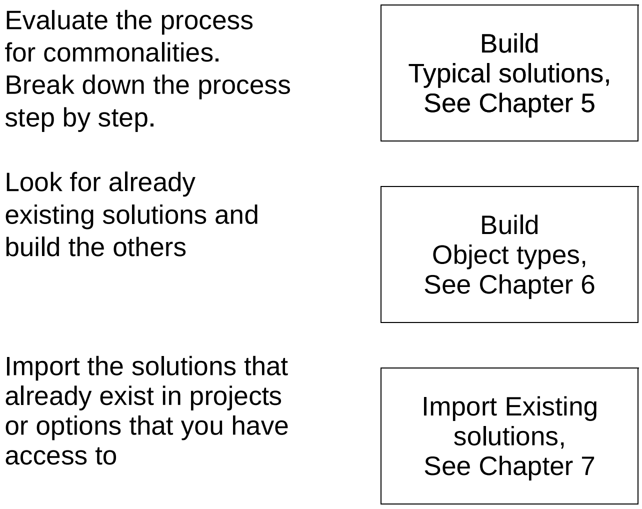

Evaluate the process for commonalities. Break down the process step by step.
Build Typical solutions, See Chapter 5
Look for already existing solutions and build the others
Build Object types, See Chapter 6
Import the solutions that already exist in projects or options that you have access to
Import Existing solutions, See Chapter 7

Figure 4-1. Creating the building blocks

As soon as you have the building blocks available you can get started. Many are provided initially when you install the graphic library options and the symbol option for the AdvaSoft 2.0 for Windows NT. Other may need to be created when you build typical solutions and new object types.

4.5 Creating the Structures and an Overview

You should start to build up the outline of the most important structures, the Functional and the Location structure, early. [1]

This is part of the engineering analyze. You gain on structuring the functionality into understandable entities and at the same time you get an overview of the full functionality and how it is represented from different views.

For similar reasons you should start to build up the overview graphic displays. Figure 4-2 gives an overview of these tasks.

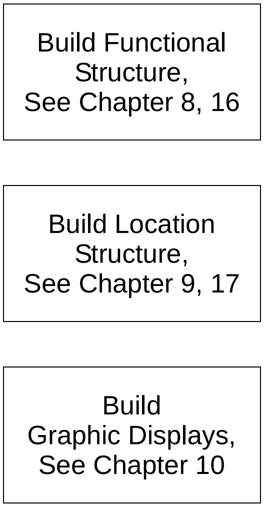

Build Functional Structure, See Chapter 8, 16
Build Location Structure, See Chapter 9, 17
Build Graphic Displays, See Chapter 10

Figure 4-2. Creating the structures and a good overview

---

[1] You need your typical solutions and all your new object types before you can create complete Functional and Location structures. Thus, you should start with an outline.

4.6 Creating the Controller Applications

The same applies for the Controller applications as for the applications described in the Advant Workplace products. You need to identify and to build typical solutions to save engineering work. You also need to structure the applications.

Most applications include functionality in both the Controllers and in the Advant Workplace Products. In this book we describe the building of the Controller applications and how they are glued together with the Advant Workplace Products in a number of separate chapters but keep in mind that it is all one solution for the total plant. See Figure 4-3 for an overview.

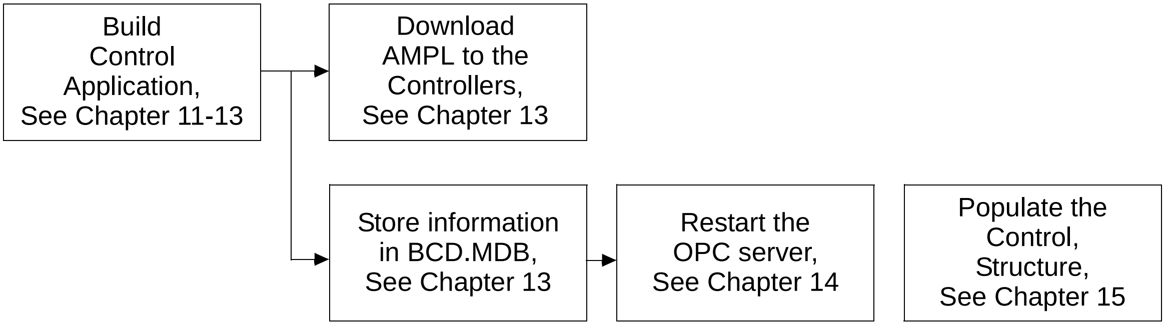

Build Control Application, See Chapter 11-13
Download AMPL to the Controllers, See Chapter 13
Store information in BCD.MDB, See Chapter 13
Restart the OPC server, See Chapter 14
Populate the Control, Structure, See Chapter 15

Figure 4-3. Creating the Controller applications

4.7 Creating the Remaining Applications

A number of applications can be created rather late in your engineering process. Examples are customizing the alarm functionality and building the history and trend applications. See Figure 4-4 for an overview.

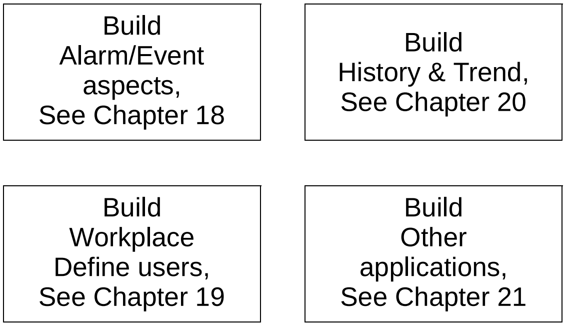

Build Alarm/Event aspects, See Chapter 18
Build History & Trend, See Chapter 20
Build Workplace Define users, See Chapter 19
Build Other applications, See Chapter 21

Figure 4-4. Creating the remaining applications

Chapter 5 Typical Solutions Overview

5.1 Introduction

The Advant Workplace Products support typical solutions as a method to decrease the engineering effort. This chapter will outline the principles and provide examples for how to use it.

The reasons to build typical solutions are many:

- You get larger building blocks. Each time you can use such a building block you decrease the total engineering time.
- Decreased engineering time due to field proven solution. A typical solution that is tested and verified does not need the same amount of testing when it is used the following times.
- You get a methodology with which you can brake down the process into smaller parts, easier to grasp and handle.

First, the basic principles for a typical solution are:

- A number of objects are collected in a small branch in (most often) the Functional structure. (A suggestion is to always create a Solution Library root object under which each respective typical solution is placed).
- Very often you create one or several new object types for part of the solution. Instances of those are included in the branch.
- Signal object are included in the branch. If they are given the right name when the typical solution is created the corresponding aspects in the Control structure will be created properly when the Control structure is created. See Chapter 15, Build the Control Structure.
- The user interaction is normally done with the top object (most logical) and this is also where you store most of the specific aspects for the typical solution.
- Graphic elements and other aspects which are included in the typical solution shall be built in such a way that they function in the instantiated typical solution.

5.2 Preparation

Create a branch called Solution Library in the Functional structure:

1. Start Advant Plant Explorer.
2. Select Functional structure.
3. Click on New Object without any selection in the structure. Select a generic type and call it Solution Library.

5.3 Our Example

The example that we are going to use is a tank with a drain system. As you can see below the Milk Reception consists of 3 such tanks in the Milk reception. In fact, there are 3 drain systems in the Milk Reception Cell, 3 in the Product Silos and 2 almost identical in the Liquid Processing Cell. The reusability is large.

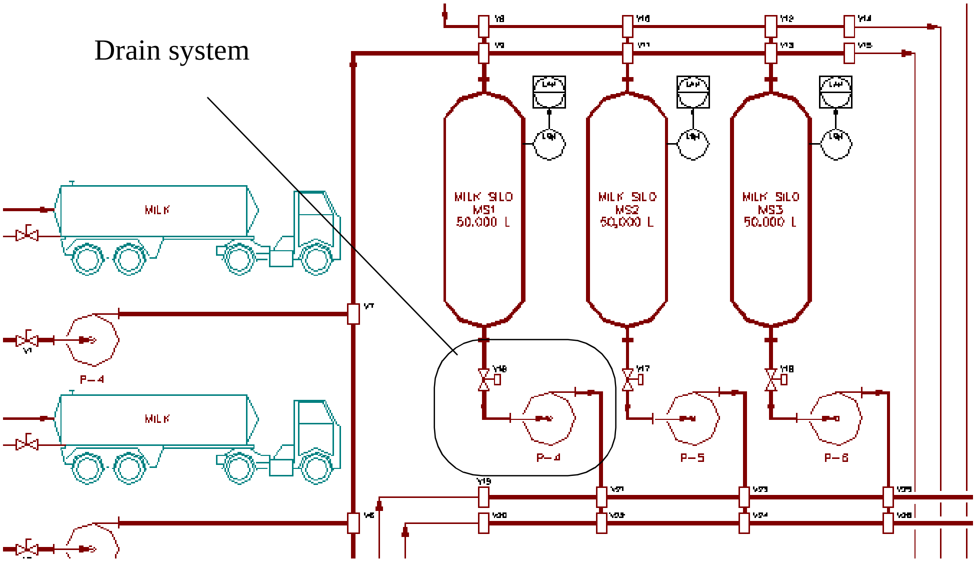
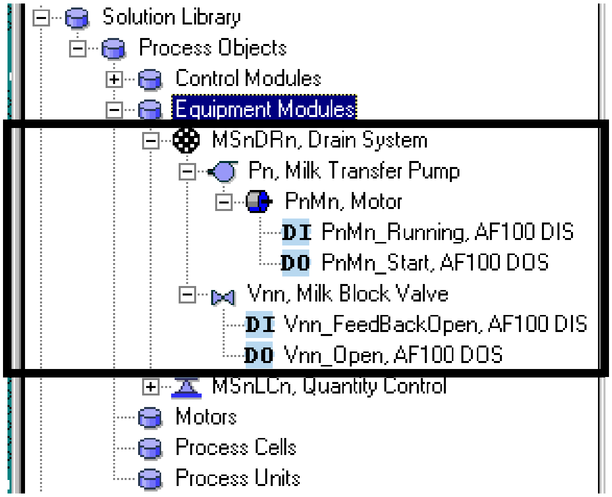

Drain system

Figure 5-1. Milk Reception Cell

Let us start with a figure that shows how the Functional structure will look like for the typical solution when you have added all the various objects to it.



- Solution Library
  - Process Objects
    - Control Modules
    - Equipment Modules
      - MSnDRn, Drain System
        - Pn, Milk Transfer Pump
          - PnMn, Motor
            - DI PnMn_Running, AF100 DIS
            - DO PnMn_Start, AF100 DOS
        - Vnn, Milk Block Valve
          - DI Vnn_FeedBackOpen, AF100 DIS
          - DO Vnn_Open, AF100 DOS
      - MSnLCn, Quantity Control
    - Motors
    - Process Cells
    - Process Units

Figure 5-2. Functional structure for our typical solution example

We want to create such a functional solution as a separate branch. Furthermore we want to create 3 graphical elements:

- a graphical element describing the Milk Transfer Pump, it will refer to the Signal objects underneath the Motor, see Figure 5-2
- a graphical element describing the Milk Block Valve, it will refer to the Signal objects underneath the Block Valve, see Figure 5-2
- a graphical element describing the whole Drain system, it will include the two first graphical elements and thus needs to be able to refer to the Milk Transfer Pump and the Milk Block Valve

5.4 Name References in a Typical Solution

A prerequisite for the graphical elements in a typical solutions to work, is that they use relative names and name paths in the expressions. Then they can be used in more than one graphic display and also be copied. We will show you below how Advant Graphics can make a relative reference to the objects underneath within the branch.

AdvantObject

If you want to create relative references, use the relative name AdvantObject instead of absolute names and name paths.

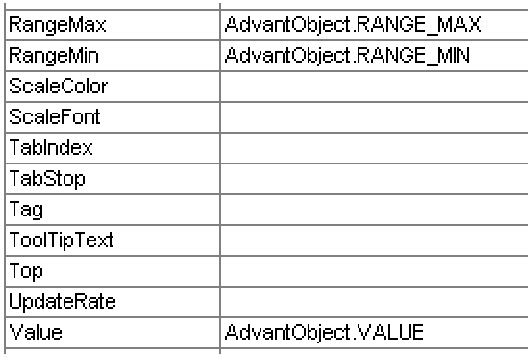
An expression with an absolute name, for example

$’TIC100_4’.VALUE * 14.2

will look like this with a relative reference

AdvantObject.VALUE * 14.2

which means that the VALUE property of the ABB Object represented by the graphic element, multiplied with 14.2, drives the property to which this expression is applied.



RangeMax	AdvantObject.RANGE_MAX
RangeMin	AdvantObject.RANGE_MIN
ScaleColor	
ScaleFont	
TabIndex	
TabStop	
Tag	
ToolTipText	
Top	
UpdateRate	
Value	AdvantObject.VALUE

Figure 5-3. Example of relative references in the Expressions window

SubPath

A SubPath is used to find a “child” object, that is another ABB Object underneath the current ABB Object. As child objects exist in most typical solutions you probably need to use the SubPaths.

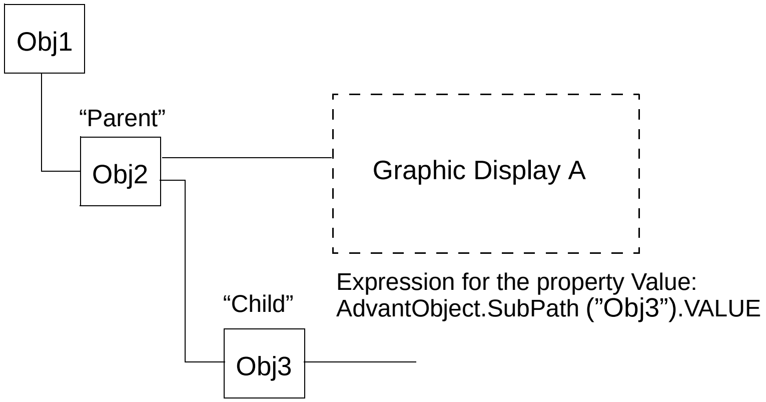

Obj1
“Parent”
Obj2
Graphic Display A
“Child”
Obj3
Expression for the property Value:
AdvantObject.SubPath (”Obj3”).VALUE

Figure 5-4. SubPath

In the subPath you can use one of these two possibilities:

- Refer to the type name of an object. You can use this if there is only one object of the specific object type underneath. You can also make combinations of several type names to make them unique, for example: Motor.AF100 DOS.
- Add a relative name aspect to the objects of the same object type and refer to the relative names.

5.5 Building the Drain system typical solution

Here is a list of steps when you build the typical solution Drain system:

1. Build the object types that are needed for the typical solution
2. Build a branch of the Functional structure that describes the solution.
3. Create a relative name aspect on the objects which are not of a unique object type and give each such object a descriptive relative name

4. Build the graphic elements
5. Copy graphic elements to their respective object types

You can also extend the solution by performing the following steps:

1. Build one or several process dialogs
2. Add other aspects that you want in the typical solution

5.5.1 Build the Object Types

See Chapter 6, Build the Object Types. To get a functioning solution you need to build the **Milk Transfer Pump** and the **Milk Block Valve** object type.

5.5.2 Build the Functional Structure for the Solution

To begin with we want to build a Solution Library branch. Do as follows:

Create new objects underneath the Solution Library object until the level where you want to place your branch. We typically group our solutions in the following layers:

- Process Objects
  - Control Modules
  - Equipment Modules
  - Motors
  - Process Cells
  - Process Units

In our example we want to create a solution for an equipment unit so we create two layers before we can create our typical solution:

- Process Objects
- Equipment Modules

Your branch now starts to resemble Figure 5-2.

Lets start to create our typical solution.

1. Start Advant Plant Explorer.
2. Select Functional structure.
3. Browse until you are placed on the Equipment Modules in the Solution Library.
4. Create a new object of the Drain System type, call it MSnDRn, Drain System (you can of course use your own name conventions). Name explanation: We know that we are going to use it for multiple Milk Storage Tanks -MSn- and it might be that there will be multiple drain system on one storage tank - DRn. With a name convention like this we will get unique names.
5. Underneath it create a Milk Transfer Pump object, Pn and a Milk Block Valve object, Vnn.
6. Underneath the Milk Transfer Pump object create a Motor object PnMn and underneath that two signal objects, an AF100 DOS called PnMn_Start and an AF100 DIS called PnMn_Running.
7. Underneath the Milk Block Valve object create an AF100 DIS object named Vnn_FeedBackOpen and an AF100 DOS object called Vnn_Open.

Verify the result with Figure 5-2.

5.5.3 Create the Relative Name Aspects

In this example we can refer to the type names. (The Milk Transfer Pump Graphical element can refer to AF100 DOS and AF100 DIS and the same applies to the Milk Block Valve graphic element). However, to show you what is needed, we use relative name references for the Signal objects underneath the Block Valve. To create them do as follows:

1. Select one of the Signal objects in the Functional structure and click on new aspect. Select **BasicPropertyAspect > BasicPropertyName > RelativeName**.
2. Click on the preview icon to get a working area for the aspect configuration and click on the relative name aspect. A configuration view is shown. Fill in the relative name (see step 3) and save the aspect.

3. Repeat for the other object. In our example the relative names are:
   - Vnn_FeedBackOpen: FeedBackOpen
   - Vnn_Open: Open

5.5.4 Create the Graphical Elements

In our example we will build three graphical elements. See Figure 5-5. You can see the Pump, the Block Valve graphical elements as well as the complete solution - the Drain system graphical element.

Let us start building them.

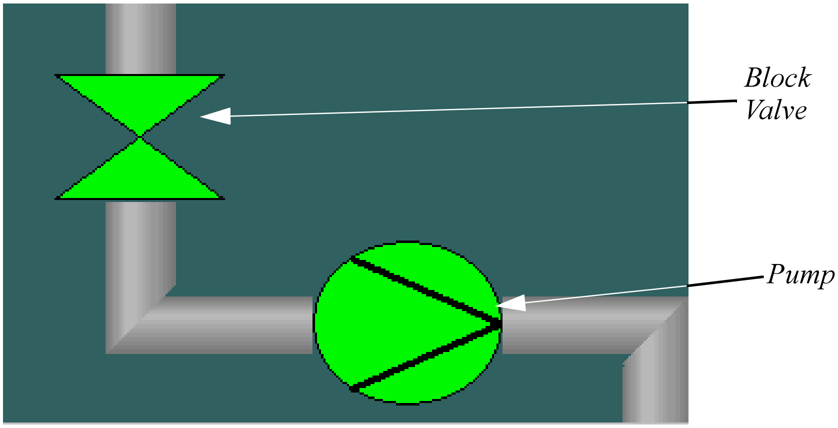
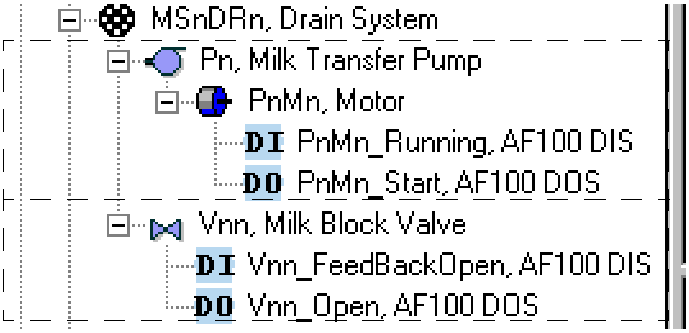


Figure 5-5. Drain system graphical element

The whole process of building a graphical element is described in detail in the *Advant Graphics User's Guide* and in the On-line help. It is a fairly large tutorial and we do not duplicate the information in here. Just follow the tutorial. Here is a brief description of the steps and a few comments though.

If you want to build your own animated graphics or import and filter photos, which are all useful additions to graphic elements/graphic displays, there are several third party products that can be used. We recommend Paint Shop Pro from JASC. Please look at http://www.jasc.com. There is a trial license for 30 days.

If you look at the solution in Figure 5-6 you see that the Pump and the Milk Block Valves are typical solutions in themselves. So the design of graphic elements is something we do for them as well as for the Drain system.



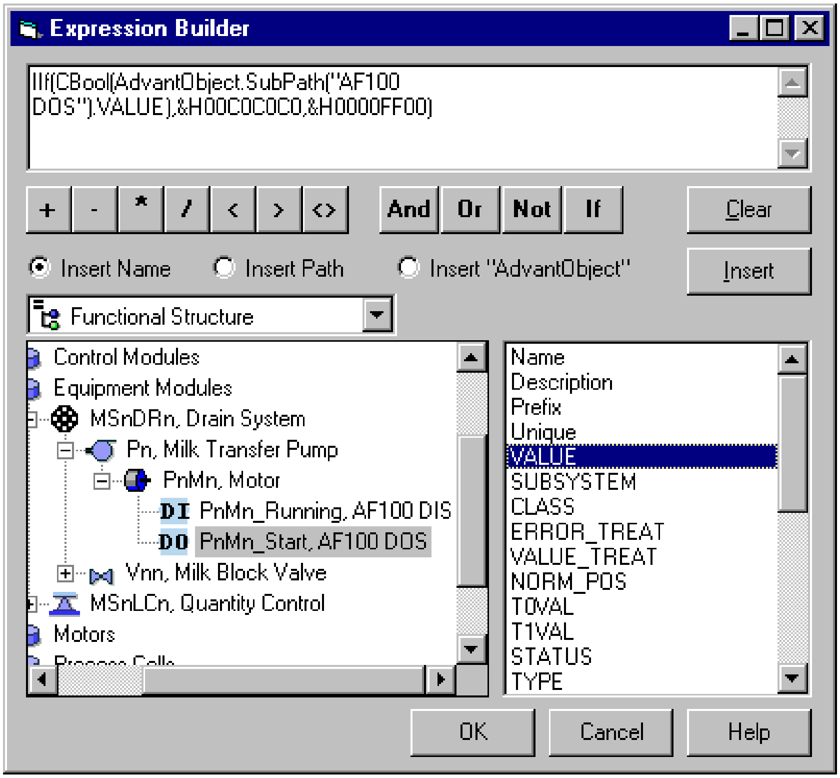
MSnDRn, Drain System
Pn, Milk Transfer Pump
PnMn, Motor
DI PnMn_Running, AF100 DIS
DO PnMn_Start, AF100 DOS
Vnn, Milk Block Valve
DI Vnn_FeedBackOpen, AF100 DIS
DO Vnn_Open, AF100 DOS

Figure 5-6. Solution tree branch

Build the graphical element for the Milk Transfer Pump, the Milk Block Valve and the Drain systems respectively. You repeat the following for all three (3) graphical elements:

- Create the Graphical element aspect and choose edit.
- Use the graphic primitives and elements you need to draw the graphical representation, see Figure 5-5. The graphic building blocks you need for this are available as part of the Advant Graphic Builder. Additional graphical primitives and elements are available if you have bought the option *Advant Graphic Library for Process Industry.*
- To open up the Expression Builder, do as follow. Select the graphical element. Select View-> Expression Window. Click in the property field that you want to effect (for example FillColor) and then click on the “=” symbol to the right to start the expression builder.

- Use the expression builder to write the data access expressions with relative and unique name references. Within the Pump Element we will refer directly to the AF100 DOS and AF100 DIS type names. This is described in the expression example from the Advant Graphic Builder. See Figure 5-7. What you see in the figure is that the color of the pump symbol is switched between grey (stopped) and green (running) depending on the state of the DOS object. We have marked the object and the attribute to show you a convenient way to find the attribute name.



Expression Builder

```
IIf(CBool(AdvantObject.SubPath("AF100
DOS").VALUE),&H00C0C0C0,&H0000FF00)
```

● Insert Name ○ Insert Path ○ Insert "AdvantObject"

Functional Structure

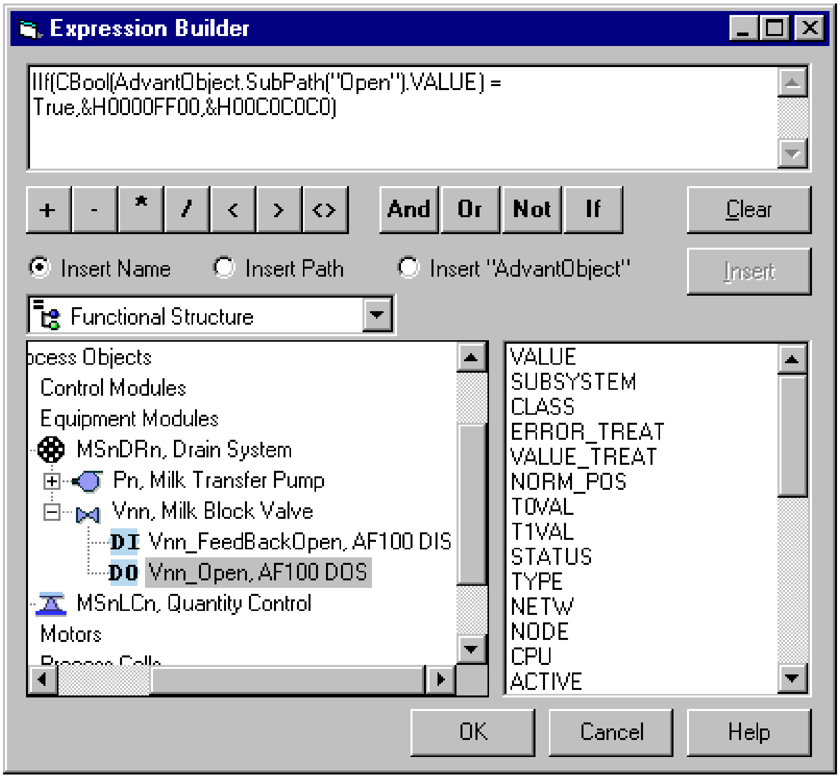
- Control Modules
- Equipment Modules
- MSnDRn, Drain System
  - Pn, Milk Transfer Pump
    - PnMn, Motor
      - DI PnMn_Running, AF100 DIS
      - DO PnMn_Start, AF100 DOS
  - Vnn, Milk Block Valve
- MSnLCn, Quantity Control
- Motors
- Process Cells

Name
Description
Prefix
Unique
VALUE
SUBSYSTEM
CLASS
ERROR_TREAT
VALUE_TREAT
NORM_POS
T0VAL
T1VAL
STATUS
TYPE

Figure 5-7. Pump graphical element - Expression Builder

- Within the Milk Block Valve we use relative names instead so we refer to the relative names created in Section 5.5.3, Create the Relative Name Aspects.

  See the Expression Builder snapshot in Figure 5-8. We switch between the colors green and grey here as well. In a more correct example we would of course use the state of both the DI and the DO to set up the appropriate color.



Expression Builder

```
IIf(CBool(AdvantObject.SubPath("Open").VALUE) =
True,&H0000FF00,&H00C0C0C0)
```

● Insert Name ○ Insert Path ○ Insert "AdvantObject"

Functional Structure

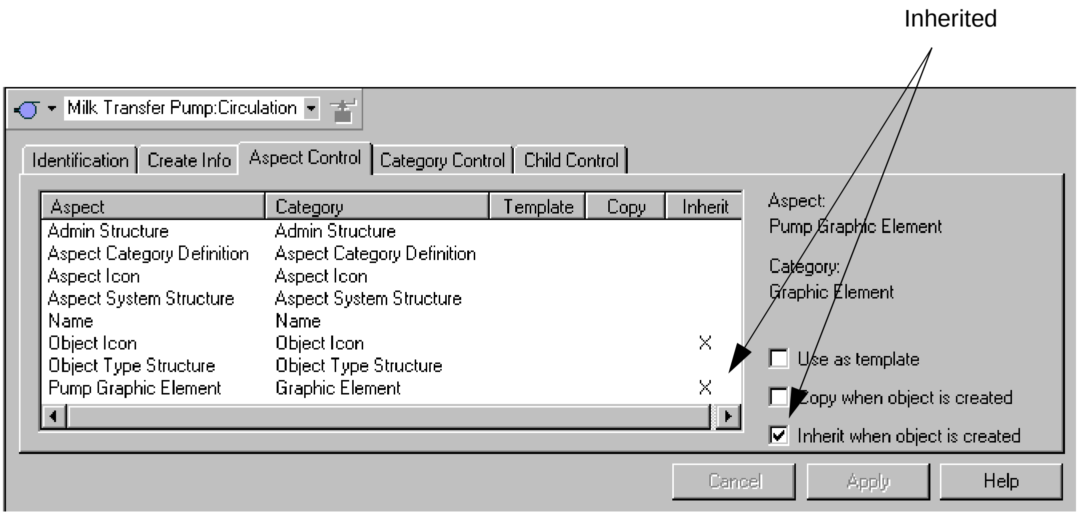
Figure 5-8. Block Valve graphical element - Expression Builder

5.5.5 Moving graphical aspects to the object types

Before we continue with the top level graphics we shall move these two graphic elements from the instances to their respective object types. The main advantage is that if you need to make any changes later on you need to do them in one place only.

So, let us copy the Graphic Element aspect to the respective object types. Use Plant Explorer and start with the Milk Transfer Pump, Pn:

1. Select the object and right click on the graphic element aspect, choose cut.
2. Select the object type structure and then browse to the object type where it shall be placed. Right click, choose Paste Special -> Aspect.
3. Click on the aspect, choose edit. Deploy and save the aspect.
4. Click on the Type Definition Aspect and choose Aspect Control, click on the Graphic Aspect and fill in the Inherited check box. See Figure 5-9.
5. Repeat for the Milk Block Valve as well.



Milk Transfer Pump:Circulation

Aspect	Category	Template	Copy	Inherit
Admin Structure	Admin Structure			
Aspect Category Definition	Aspect Category Definition			
Aspect Icon	Aspect Icon			
Aspect System Structure	Aspect System Structure			
Name	Name			
Object Icon	Object Icon			X
Object Type Structure	Object Type Structure			
Pump Graphic Element	Graphic Element			X

Aspect:
Pump Graphic Element

Category:
Graphic Element

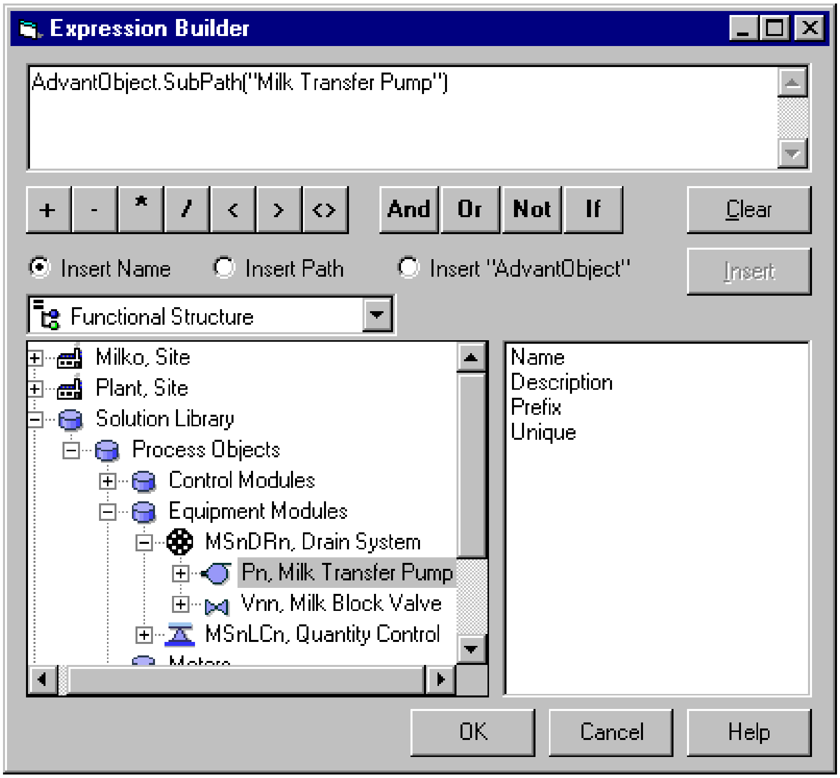
☐ Use as template
☐ Copy when object is created
☑ Inherit when object is created

Figure 5-9. Making the Pump Graphic Element inherited

Now you are ready to create the Drain system graphic element as presented in Figure 5-5. You include the Milk Transfer Pump and the Milk Block Valve graphic elements in your Drain system graphic element by using the Graphic Element Browser:

- Browse to the Pn object and select the graphic element Pump Element. Place it on the Horizontal pipe. (You started by constructing the drain system pipes.)
- Browse to the Vnn object where the Valve graphic element is found. Place it onto the Vertical pipe.

The next step is to write the expressions that makes the whole graphic element function together and support copying of the whole typical solution.



```
AdvantObject.SubPath("Milk Transfer Pump")
```

● Insert Name ○ Insert Path ○ Insert "AdvantObject"

Functional Structure

Figure 5-10. Drain system graphic element, pump reference

The Drain system graphic element can refer to the Milk Transfer Pump and to the Milk Block Valve respectively as seen in Figure 5-10. What you do in the expression builder is to fill in the property Advant Object for the pump and for the block valve symbol respectively. Do as follows:

1. Click on the Pump Symbol
2. Select View Expression Window, start the expression builder for the AdvantObject property.
3. Fill in the subPath reference as you can see in Figure 5-10 and click OK.
4. Select the Block Valve symbol by clicking on it. You will get the Block Valve presented in the expression window. Start the Expression Builder and fill in a subPath reference to the Milk Block Valve.

5.5.5.1 Building a Process Dialog

The whole process of building a process dialog is described in detail in the *Advant Graphics User's Guide* and in the on-line help. A few notes though:

- A Process dialog is built up by a number of tabs.
- Create the additional tabs you need (note that you can use the graphical element built in the previous section as one of the tabs).
- Create the Process Dialog aspect using the configuration view (it can be automatically or manually configured).

A process dialog can be achieved in two ways:

- Create and configure an aspect of the category Process Dialog (the recommended way).
- Create a graphic element, which can also be used as a dialog, separately.

NOTE

Do not forget to make the name references relative and the use of the subpath functionality.

5.5.6 Adding Additional Aspects to the Typical Solution

5.5.6.1 Alarm List Aspect

If you want an alarm and event list aspect for the objects that are included in the typical solution, do as follows:

1. Create an Alarm List aspect on the “top” object of your typical solution.
2. Select the Alarm List Config View.
3. Check the “Object and descendents in structure” checkbox and select the Functional structure.

Chapter 18, Building Alarm and Event Applications includes a tutorial on how to do this for a whole process cell and the *Advant Alarms User’s Guide* contains further details.

As the alarm list aspect just uses the reference to the structure you configured it towards (functional) it is independent of the object names and subsequently can be copied without being modified.

In our example we do it on the Drain System object.

5.5.6.2 Trend Aspect

If your typical solution contains signal objects that are relevant to present in a common Trend presentation you should create such an aspect. You just refer to the name and the relevant attribute for each one of the traced objects. Trend aspects do not support relative addressing so you have to edit the Trend aspect after copying it.

In our solution we only have digital in and output signals which are not so interesting to trend, thus we do not add any trend aspect. Each one of the signal objects already have an inherited Trend aspect.

5.6 Using the Typical Solution

Now it’s time to use the typical solution. We want to create the 3 Silo tanks MS1, MS2 and MS3 together with their Drain system and Level Control. This is simple.

1. Open up the Advant Plant Explorer, browse the Functional structure to the Milk Reception Cell object MR1.
2. Click on the Create new object icon and create the Milk Silo object MS1.
3. Browse to the typical solution object MSnDRn, right click and select Copy from the context menu.
4. Click on the MS1 Silo object and then use the Context menu to select Paste. The MSnDRn and it’s branch of objects and aspects are placed underneath the MS1 object.
5. Click on the MSnDRn object. Open the preview area and click on the name aspect. Change the name to MS1DR1.
6. Repeat the name change for all objects according to the following table:

Table 5-1. Name changes

Typical name	Final name
MSnDRn	MS1DR1
Pn	P4
PnMn	P4M1
PnMn_Running	P4M1_Running
PnMn_Start	P4M1_Start
Vnn	V16
Vnn_FeedBackOpen	V16_FeedBackOpen
Vnn_Open	V16_Open

7. Select the Drain system graphic aspect, choose edit, then deploy and save it.

We will get a solution that looks like Figure 5-11.

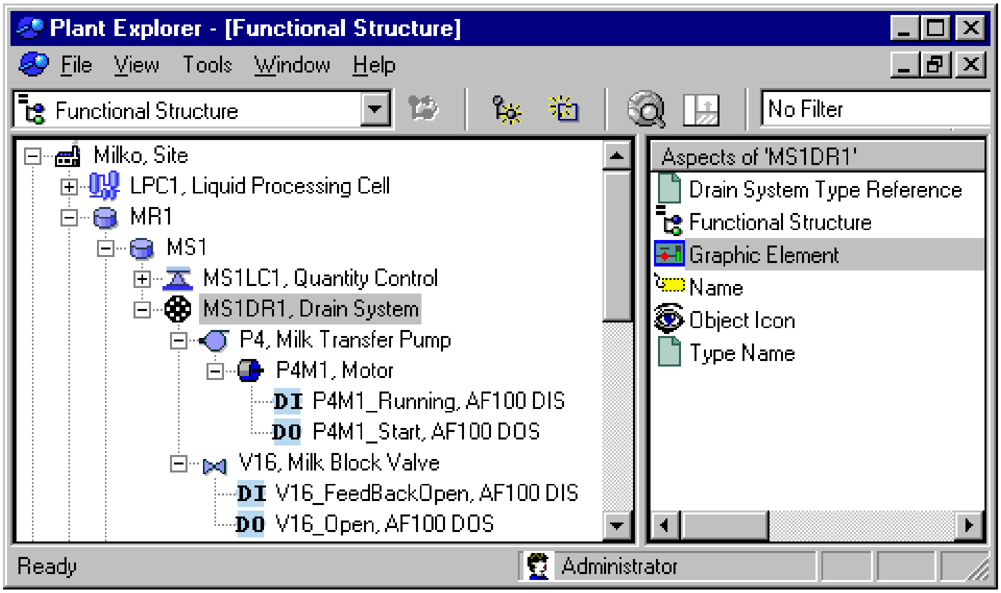


Figure 5-11. Result after a copy of the typical solution

The solution will start to function when the Control structure is populated as described in Chapter 15, Build the Control Structure.

8. Finally, repeat this for the MS2 and MS3 Silo.
9. Do the same steps for all 3 Silos when adding the Quantity Control typical solution.

In Chapter 10, Build Graphic Displays we will show you how to build a graphical display for the 3 silos using the Drain system graphical element. See Figure 5-12 for a preview of an intermediate step.

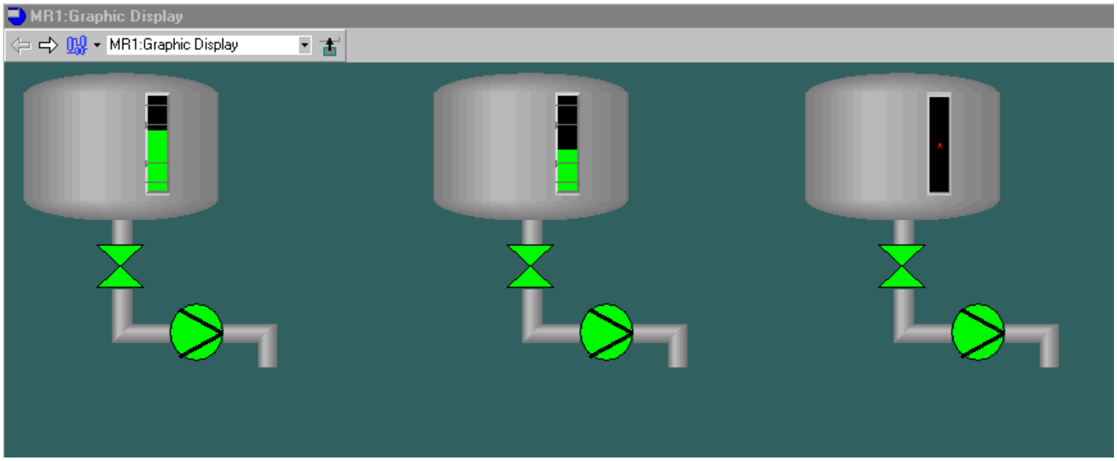


Figure 5-12. The MR1 Graphic Display

5.7 Limitations in AdvaSoft 2.0 for Windows NT

The following limitations exist in this release:

- You should only copy a typical solution within the same structure, thus the solution library branch in the Functional structure, as the structure aspects are not copied.
- Import and export does not handle dependencies. Thus you must make sure to export and import all relevant object types, objects and aspects yourself. See Chapter 7, Export/Import of Existing Solutions for details.

Chapter 6 Build the Object Types

6.1 Introduction

The reason to create new object types are similar to why you build typical solutions. It makes it possible for you to get dedicated and specialized building blocks, thus decreasing the engineering effort. This chapter describes the essentials around object types. See Section 2.6, ABB Object Type Concept for more information why to use object types and what they give you. See also Section 6.5, Object types provided with the release for more information about existing object types.

In our examples we will use a number of new object types that are not included as part of the product release. We will build the Milk Transfer Object Type as an example.

The first action is to create the new object type:

1. Select the Object Type Structure. Browse to a suitable place in the structure to place it in.
2. Select **New Object** on the Object tree view context menu.
3. Select **Object Type**, which is the only Object available for this operation.
4. Enter a name, in this case Milk Transfer Pump and click **Create**.

Secondly, create and configure an Icon as described in Section 6.2, Creating an Icon Aspect.

Third, add any other aspects that you want to include in the object type. Remember, building an object type is just like creating and customizing any other object but the object type will be used as a template by all it´s instances.

Fourth, configure the Object Type according to your needs. See Section 6.4, Configuring the Object Type Definition.

6.2 Creating an Icon Aspect

If you create a new object type you should also select your own Icon for it.
In this document we do not describe how you create icons using one of the many possible graphical editors, we just describe how you use an available icon. We

recommend that you use AX Icons from Axialis to build your Icons, see http://www.axialis.com/axicons/. It is not an expensive product (currently in the range of 15 USD) and it has a trial license for 30 days so you can start using it immediately.

To add an Icon aspect and an Icon from the Advant Plant Explorer:

1. Select the new object type and create an Icon aspect
2. Right click on the Icon aspect and select the Config View.

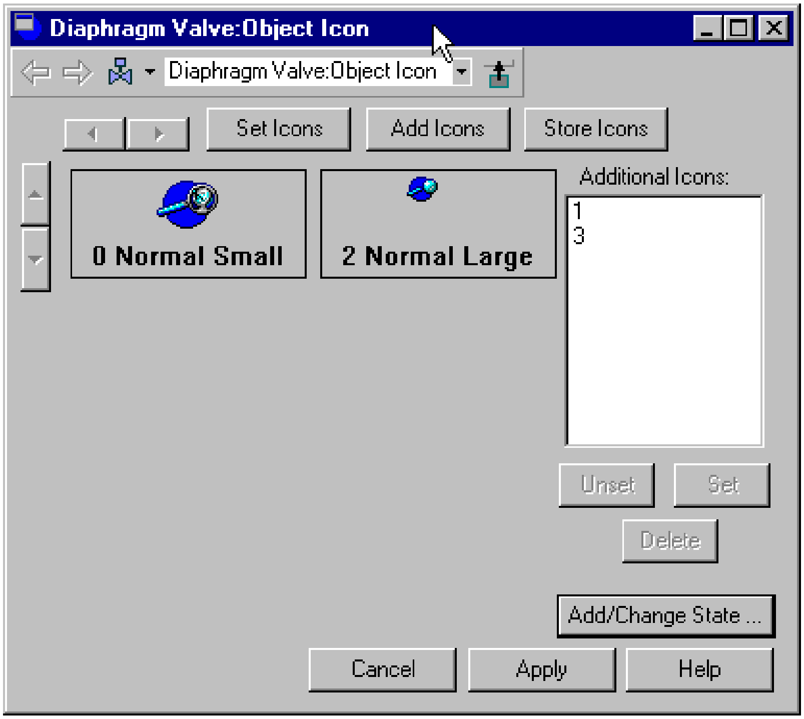


Figure 6-1. Selecting an Icon

3. Click on the **Set Icons** button.
4. Browse the file system for the Icon you have built and select it.
5. Click on **Apply** in the Icon Configuration view.

6.3 Creating other aspects

Now you can build the additional aspects that you want to complement the object type with. Graphical aspects, process dialogs, web aspects, windows application aspects, your own aspects etc.

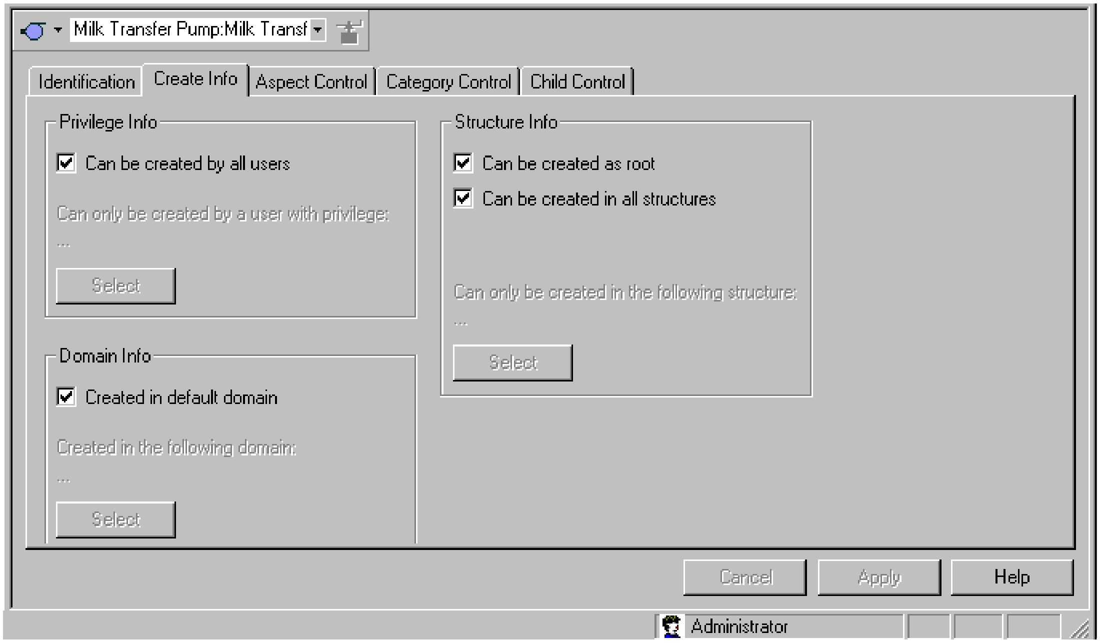
You can add or modify an aspect of an object type later. Please remember though that if it is an aspect that is copied with each new instance or used as a template, the changes will not propagate to the already created instances.

6.4 Configuring the Object Type Definition

Select the Milk Transfer Pump Type Definition aspect and configure the following things, see also the theory behind in Section 2.6, ABB Object Type Concept and the detailed description in the *Advant Plant Explorer User's Guide*. This book will not describe all details and configuration fields.

6.4.1 Privilege and Structure Information

The first thing you describe is which users that may create instances of this object type and where it can be created, see Figure 6-2. We do not make any limitations in this example.



☑ Can be created by all users

Can only be created by a user with privilege:
...

☑ Created in default domain

Created in the following domain:
...

☑ Can be created as root

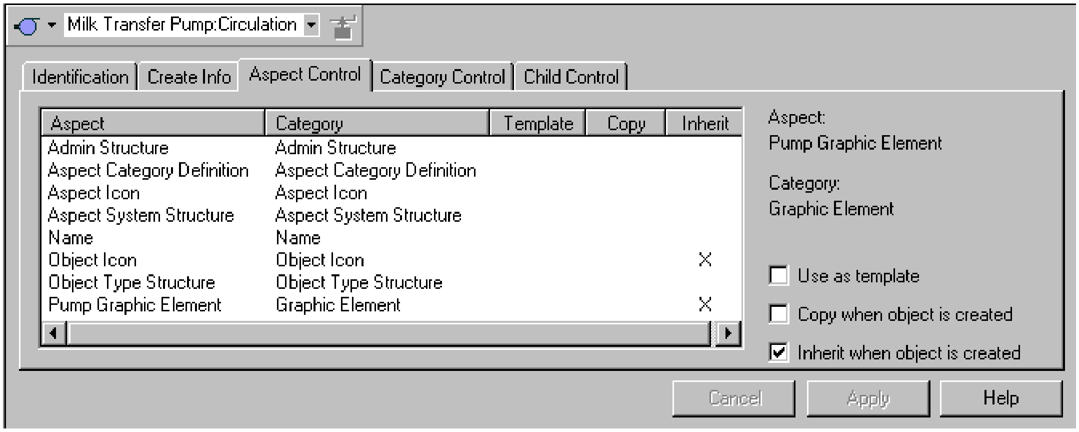
☑ Can be created in all structures

Can only be created in the following structure:
...

Figure 6-2. Create information (control)

6.4.2 Inheritance

The typical usage of inheritance is that you want all objects of an object type to always have certain aspects, for example the icon aspect. You want all objects of that type to be represented by the same icon. If you want to change the aspect (Icon in this case) you do it in one place and all the object instances are immediately updated. Compare this with the aspects that are created for each instance. In the Aspect Control tab you specify which aspects to inherit (Object Icon and Pump Graphic Element), see Figure 6-3.



Aspect	Category	Template	Copy	Inherit
Admin Structure	Admin Structure			
Aspect Category Definition	Aspect Category Definition			
Aspect Icon	Aspect Icon			
Aspect System Structure	Aspect System Structure			
Name	Name			
Object Icon	Object Icon			X
Object Type Structure	Object Type Structure			
Pump Graphic Element	Graphic Element			X

Aspect:
Pump Graphic Element

Category:
Graphic Element

☐ Use as template
☐ Copy when object is created
☑ Inherit when object is created

Figure 6-3. Aspect Control

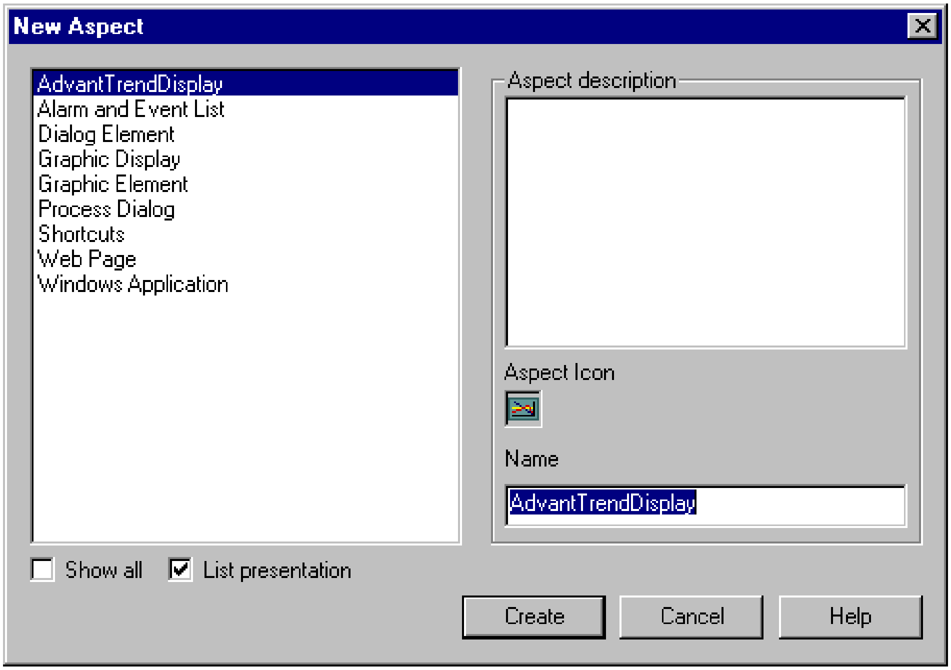
We have no aspects that we want to Copy by default or that we want to specify as template.

All aspects which provide a view for editing, can be modified later. Remember that if you want to change the aspect for all instances, the only way to do it for aspects that are not inherited is to modify them one by one for each instance.

6.4.3 Proposed and Allowed Aspect Categories

When you create a new aspect for an object the new aspect dialog will propose the aspects included in the “proposed list” that you have the privilege to create.

This is a way to customize the engineering and simplify the task of quickly selecting the new aspects at creation. In our example we want the new aspect dialog as seen in Figure 6-4.



Name: AdvantTrendDisplay

☐ Show all ☑ List presentation

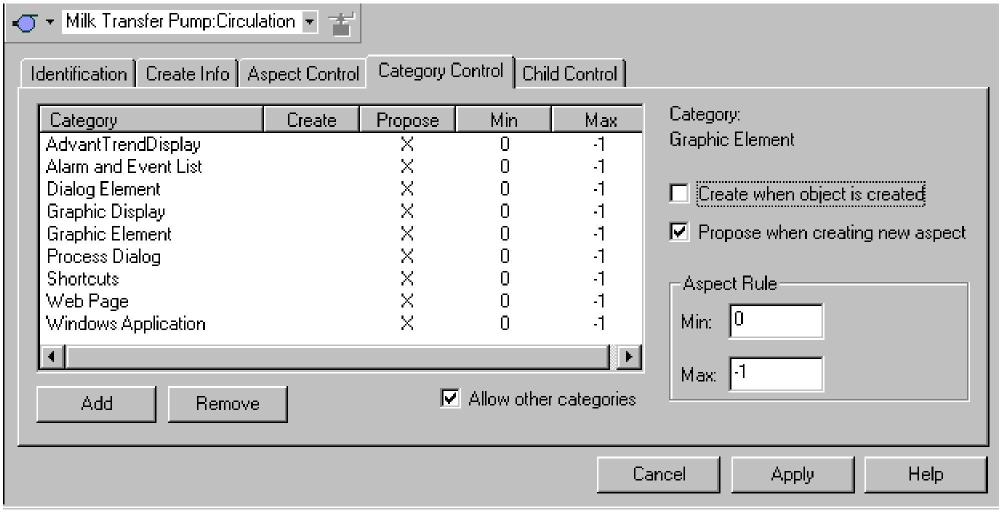
Figure 6-4. New aspect dialog

In Figure 6-5 you can see the proposed list that is required.

The new aspect dialog provides other views to look at if you want to create aspects outside the “proposed list”. If you have checked “Allow other categories” in the Category Control tab (Figure 6-5) your privileges determine which aspects that are proposed in that case.

6.4.4 Aspects that are Automatically Created

When you create an object instance, a number of pre-defined aspects are automatically created according to the object type definition. You configure this definition according to Figure 6-5.



Category	Create	Propose	Min	Max
AdvantTrendDisplay		X	0	-1
Alarm and Event List		X	0	-1
Dialog Element		X	0	-1
Graphic Display		X	0	-1
Graphic Element		X	0	-1
Process Dialog		X	0	-1
Shortcuts		X	0	-1
Web Page		X	0	-1
Windows Application		X	0	-1

Category:
Graphic Element

☐ Create when object is created

☑ Propose when creating new aspect

Aspect Rule
Min: 0
Max: -1

☑ Allow other categories

Figure 6-5. Category Control

In our example there are no aspects, except the inherited ones, to create when the object is created. We propose a list of the most plausible aspects to create.

As we do not want to restrict the aspect categories that should be allowed for the object instances we check the box “Allow other categories”.

We set Max to -1 meaning that there are no limitations on the number of aspects.

6.4.5 Proposed and Allowed Child Objects

When you create child objects underneath instances of an object type the new object dialog will propose a “proposed list” of object types.This the way we want the dialog for a Milk Transfer Pump object to look like.

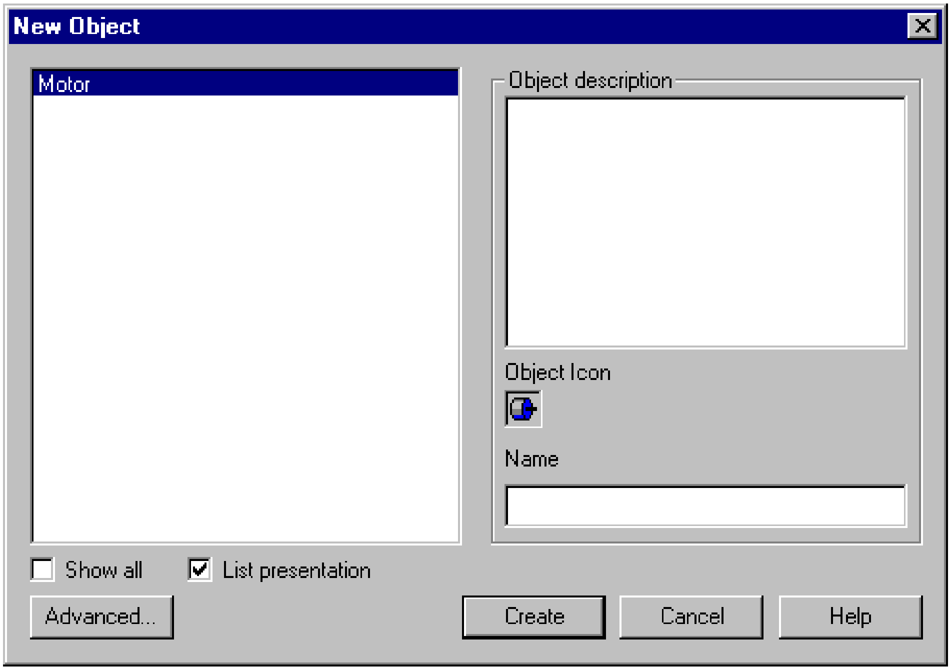
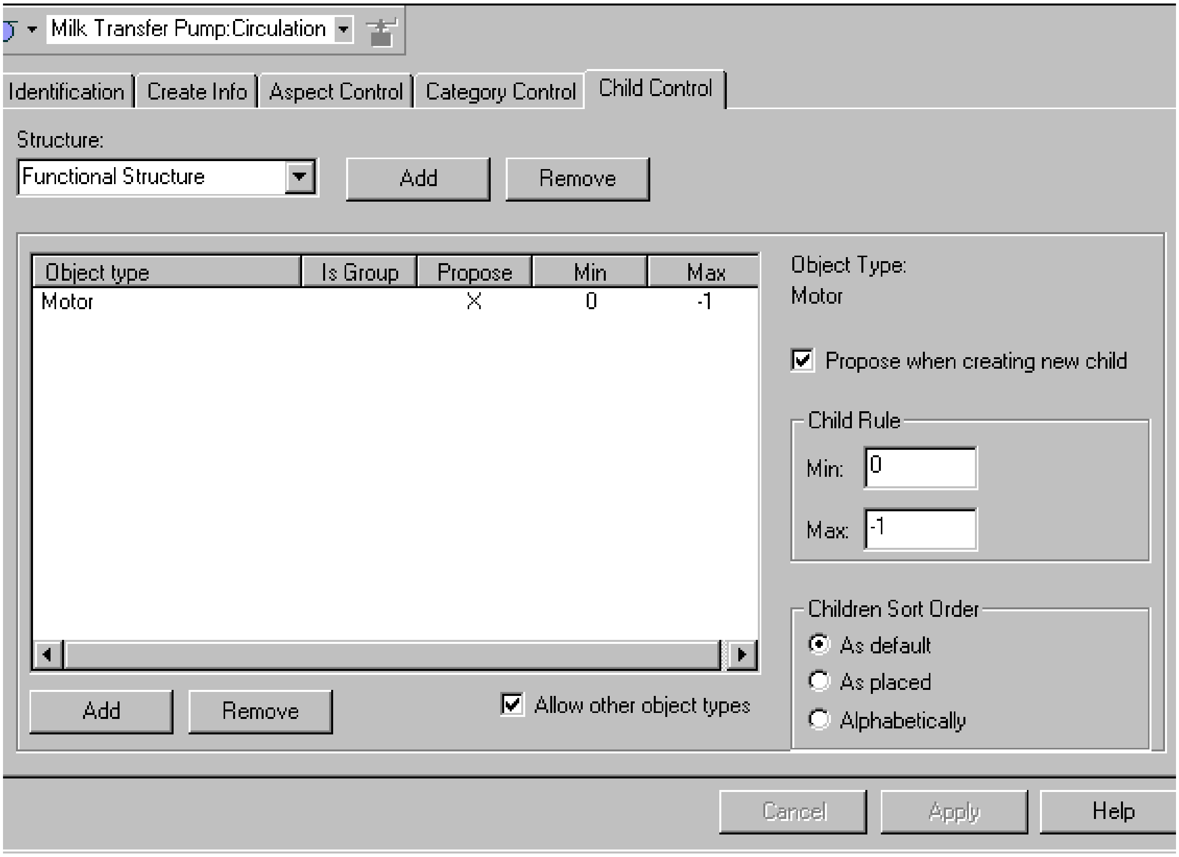


Figure 6-6. New object dialog showing a proposed list

This is a way to customize the engineering and simplify the task of quickly selecting the new object type to instantiate. See Figure 6-7 for the configuration of the Milk Transfer Pump object type.

The new object dialog provides other views to look at if you want to override the “proposed list” and create objects of other types. If the “Allow other object types” is not checked, the object types are restricted to the types in the Child Control list. If the “Allow other object types” is checked, any object type that you have privileges to create will be proposed.

In our examples we check “Allow other object types”, see Fig Figure 6-7.



Milk Transfer Pump:Circulation

Structure: Functional Structure

Object type	Is Group	Propose	Min	Max
Motor		X	0	-1

Object Type:
Motor

☑ Propose when creating new child

Child Rule
Min: 0
Max: -1

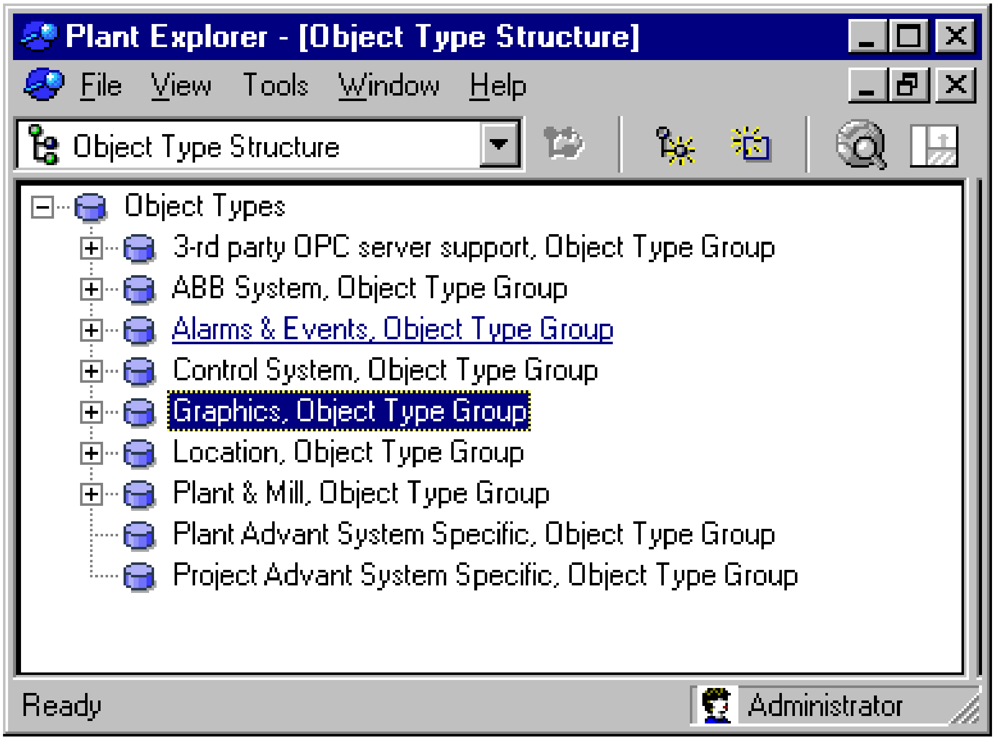
Children Sort Order
● As default
○ As placed
○ Alphabetically

☑ Allow other object types

Figure 6-7. Child control

6.5 Object types provided with the release

The existing object types are provided in the object type structure according to Figure 6-8.


Plant Explorer - [Object Type Structure]
File View Tools Window Help
Object Type Structure
Ready
Administrator

- Object Types
  - 3-rd party OPC server support, Object Type Group
  - ABB System, Object Type Group
  - Alarms & Events, Object Type Group
  - Control System, Object Type Group
  - Graphics, Object Type Group
  - Location, Object Type Group
  - Plant & Mill, Object Type Group
  - Plant Advant System Specific, Object Type Group
  - Project Advant System Specific, Object Type Group

Figure 6-8. Object type structure

The object types are organized as follows.

6.5.1 3-rd party OPC server support

This group is currently only for internal use within ABB.

6.5.2 ABB System

This group is currently only for internal use within ABB.

6.5.3 Alarms and Events

This group is currently only for internal use within ABB.

6.5.4 Control System

Here you find the network, controller and signal object types.This release contains the object types for the Advant Fieldbus 100 (Advant Controller 100 series). See Chapter 15, Build the Control Structure for details.

6.5.5 Graphics

This group is currently only for internal use within ABB.

6.5.6 Location

This is a small group, currently consisting of two object types, building and room. See Chapter 9, Build the Location Structure for details.

6.5.7 Plant & Mill

Here are all the object types describing functional entities in the plant. Chapter 8, Build the Functional Structure contains a more detailed description.

6.5.8 Plant Advant System Specific

Here the different ABB companies can place their branch specific object types.

6.5.9 Project Advant System specific

Here you can place your own, project specific, object types.

Chapter 7 Export/Import of Existing Solutions

7.1 Export/Import function

The AdvaSoft 2.0 for Windows NT is based upon a database called the aspect directory. All the data in the system is based on aspects or referred to from aspects.

The Export and Import tool allows you to export data from the aspect directory, in the form of objects and aspects, into so called archive files in the file system.

The Export and Import tool also allows you to store and restore objects and aspects into the Aspect Directory from corresponding export files (archives).

An archive is a structured storage file where the extracted data of objects and aspects is stored.

The tool's user interface allows you to either drag and drop or to use menu selections to perform export and import.

7.2 The Milko Chemical example

We have built our Functional structure down to a level above devices such as Valves, Motors, Pumps etc. We have also seen in Chapter 5, Typical Solutions Overview how we built up a typical solution for a Drain system. Our next step is to import some existing solutions from a previous project:

- A number of object types that are not included as standard
- A Quantity Control solution

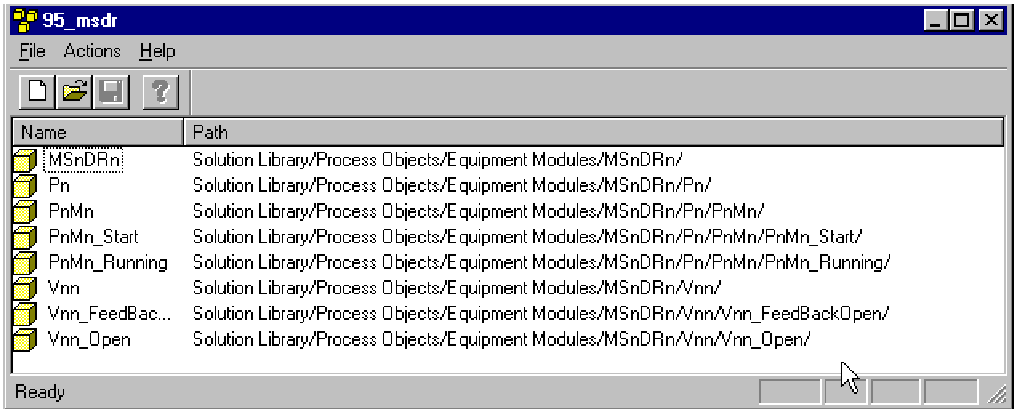
The Export/Import function in AdvaSoft 2.0 for Windows NT does not handle dependencies automatically.

When you copy (export) ABB Objects, all their aspects and aspect data are copied. However, you will not automatically get all ABB Objects and aspects that are referred to by any of those aspects, included. Those you have to identify and add to the export yourself.

7.3 Exporting the Drain system Solution

To give you an example of how to export a typical solution, here is how you do to export the Drain system. To begin with you export the none standard object types - in this case the Milk Transfer Pump - and then the solution.

1. Start the Export/Import utility from the Start menu (**Start Program> Advant > Export Import**).
2. Start Advant Plant Explorer and select the Object Type structure
3. Drag the Milk Transfer Pump object type to the Export/Import tool and store it in a file called 75_msdr.
4. Select the Functional structure.
5. Browse to the Drain system solution.
6. Drag the Drain system from Advant Plant Explorer to the Export/Import tool.



Name	Path
MSnDRn	Solution Library/Process Objects/Equipment Modules/MSnDRn/
Pn	Solution Library/Process Objects/Equipment Modules/MSnDRn/Pn/
PnMn	Solution Library/Process Objects/Equipment Modules/MSnDRn/Pn/PnMn/
PnMn_Start	Solution Library/Process Objects/Equipment Modules/MSnDRn/Pn/PnMn/PnMn_Start/
PnMn_Running	Solution Library/Process Objects/Equipment Modules/MSnDRn/Pn/PnMn/PnMn_Running/
Vnn	Solution Library/Process Objects/Equipment Modules/MSnDRn/Vnn/
Vnn_FeedBac...	Solution Library/Process Objects/Equipment Modules/MSnDRn/Vnn/Vnn_FeedBackOpen/
Vnn_Open	Solution Library/Process Objects/Equipment Modules/MSnDRn/Vnn/Vnn_Open/

Figure 7-1. Exporting a typical solution.

All objects below the dragged object (in the current structure) are automatically included.

7. Store the exported data to a file - 95_msdr.

Naming various exported files with a number first helps you to import the files in the right order. Normally object types first, solution second. Object types must be exported to a different file than solutions (instances) of those object types.

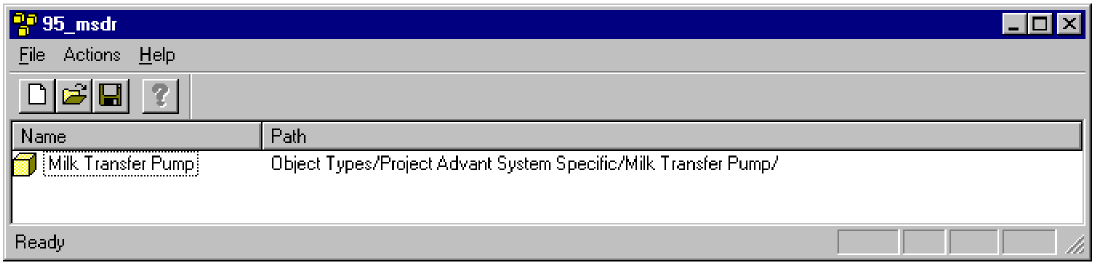
7.4 Importing Existing Solutions

To import existing solutions:

1. Start the Export/Import utility from the Start menu (**Start Program> Advant > Export Import**).
2. Click on the Open icon and browse until you have found the file with the typical solution. Select it and click **OK**.
3. Extract it (**Actions > Extract**) into the Functional structure.
4. Verify that the imported solution is visible under Solution Library in the Functional structure and that any Object types show up in the Object Type structure.

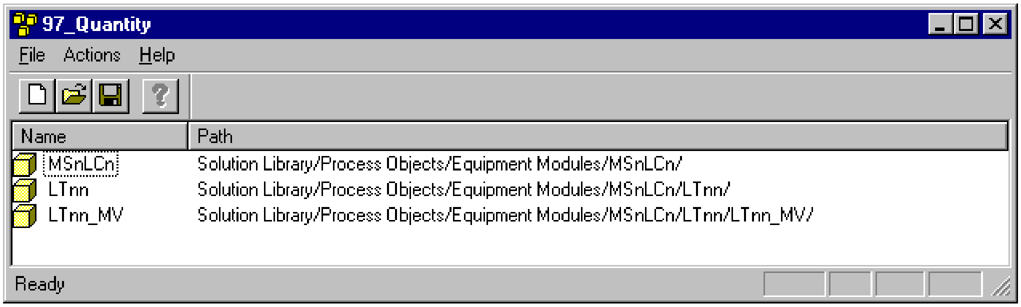
Down below in Figure 7-2 is an example of how the export file from Section 7.2, The Milko Chemical example, looks like.

Further down, Figure 7-3, you can se the import file used to import the Quantity Control solution.



Name	Path
Milk Transfer Pump	Object Types/Project Advant System Specific/Milk Transfer Pump/

Figure 7-2. Example of import file - Milk Transfer Pump



Name	Path
MSnLCn	Solution Library/Process Objects/Equipment Modules/MSnLCn/
LTnn	Solution Library/Process Objects/Equipment Modules/MSnLCn/LTnn/
LTnn_MV	Solution Library/Process Objects/Equipment Modules/MSnLCn/LTnn/LTnn_MV/

Figure 7-3. Example of Import file - Quantity Control

Chapter 8 Build the Functional Structure

8.1 Creation of ABB Objects in the Functional Structure

You should start to build up the outline of the most important structures, the Functional and the Location structure, early.

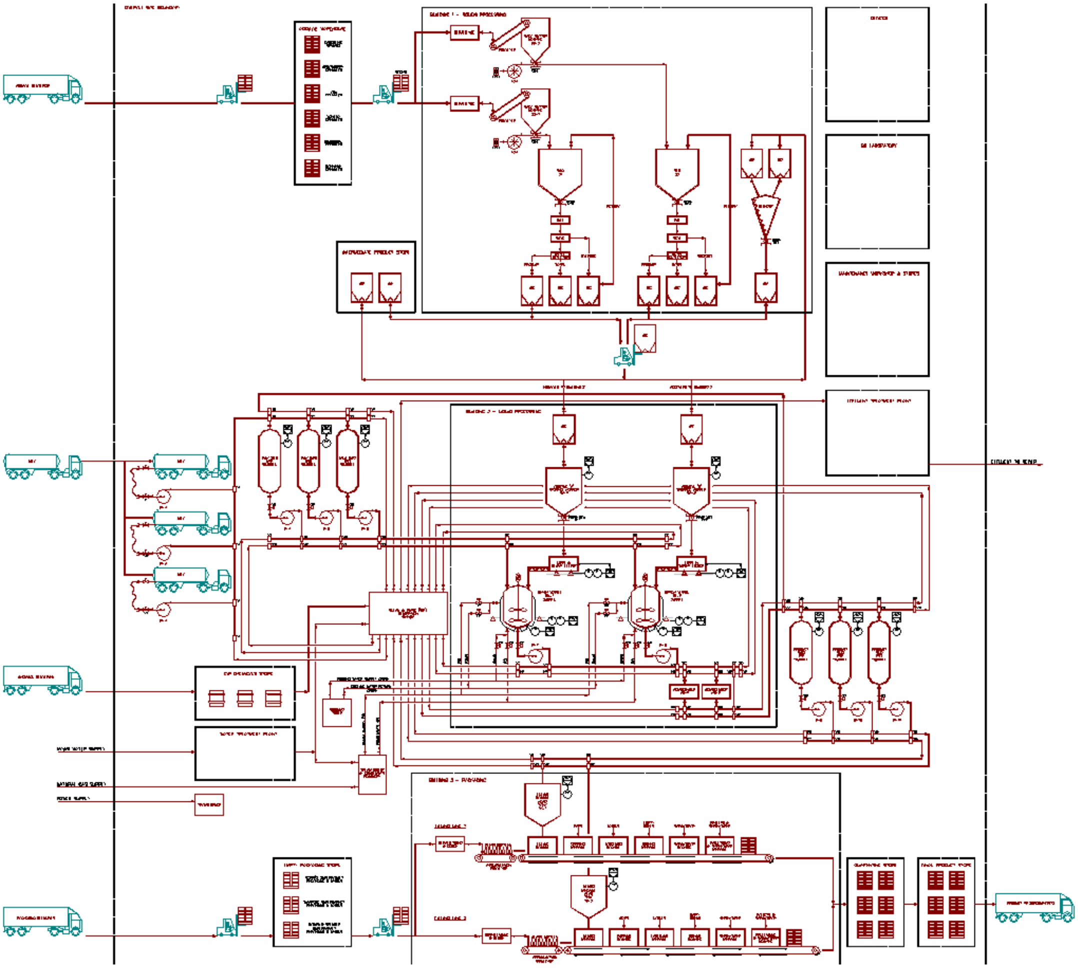


Figure 8-1. Plant overview over Milko Milkshakes - CAD drawing

You gain on structuring the functionality into understandable entities and at the same time you get an overview of the full functionality and how it is represented from different views. Use the Overview picture of the Milko Chemical, see Figure 8-1.

The plant functions to be placed in the first level below the project are the functions of the main plant sections (subplants) which, in our example, are:

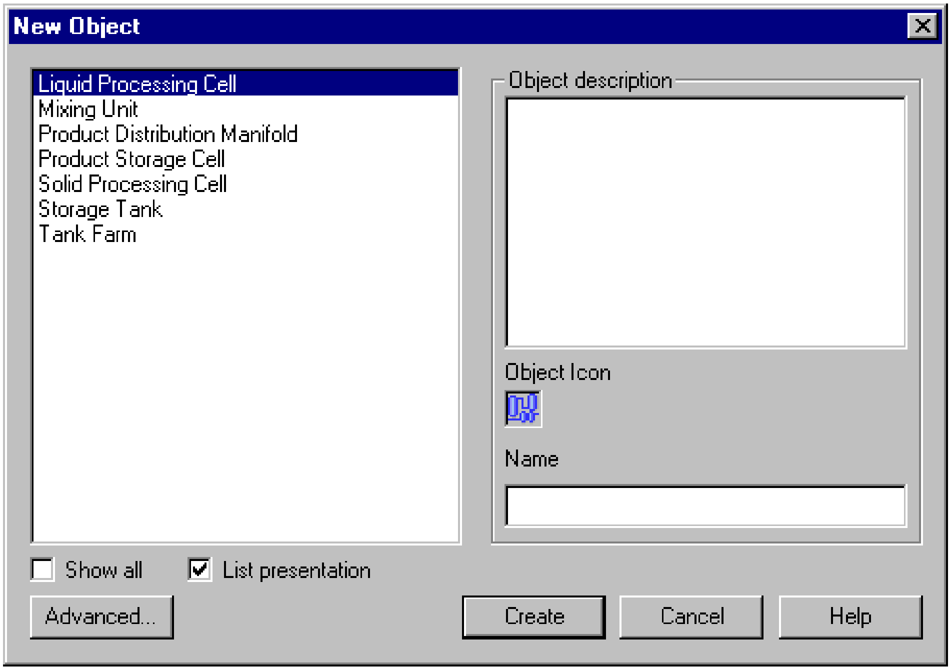
- Liquid Processing Cell
- Product Storage Cell
- SP1 Solid Processing Cell
- Milk Reception Cell

In Chapter 5, Typical Solutions Overview, we started to analyze what to do. As a result we now have a number of building blocks, object types and typical solutions, that increase the engineering speed significantly.

The Functional structure (except your solution library) is still empty except for the default structure provided by the product. Create a top object and the other top structure objects using the steps described below. We describe this in detail so you can get started.

To create a new ABB Object using the toolbar buttons, follow these steps:

1. Start Advant Plant Explorer and select the Functional structure.
2. The first time you just click on the Functional structure node before you do the next step. For the following objects you click an ABB Object, to which you want to add a new child Object or a typical solution.
3. Use the **New Object** button to start the New Object dialog.



☐ Show all
☑ List presentation

Figure 8-2. New object dialog

In this dialog you can select the Object type for the new Object, as well as give a description of the Object and name it. Use the Process objects suitable for this industry, you find them in Section 8.4, Process Object Types to be Used in the Functional Structure.

4. Click **Create**. The new ABB Object is created as a child Object to the Object you selected. The parent-child relationship is represented by the +/- signs, showing that an Object has a child attached to it. If you click a structure node and create a new Object, the Object will become a child to the node pointed at.

5. Repeat step 3-5 for all the main building blocks until all the process objects in the Functional structure are in place.You can also add Signal objects in this stage. As long as they have exactly the same name as the object in the Controller they will be used in the population of the Control structure (uploading) as described in Chapter 15, Build the Control Structure instead of creating a new object.
6. You should also use typical solutions that you copy from the solution library as described in Chapter 5, Typical Solutions Overview to improve the speed of your work.
7. During your work you will get a result similar to the structure in Figure 8-3.

To learn about all the functionality and details with regards to object creation please see the *Advant Plant Explorer User's Guide*. The steps above are somewhat simplified and do not describe all the features you can use.

8.2 Assigning Aspects (application data) to ABB Objects

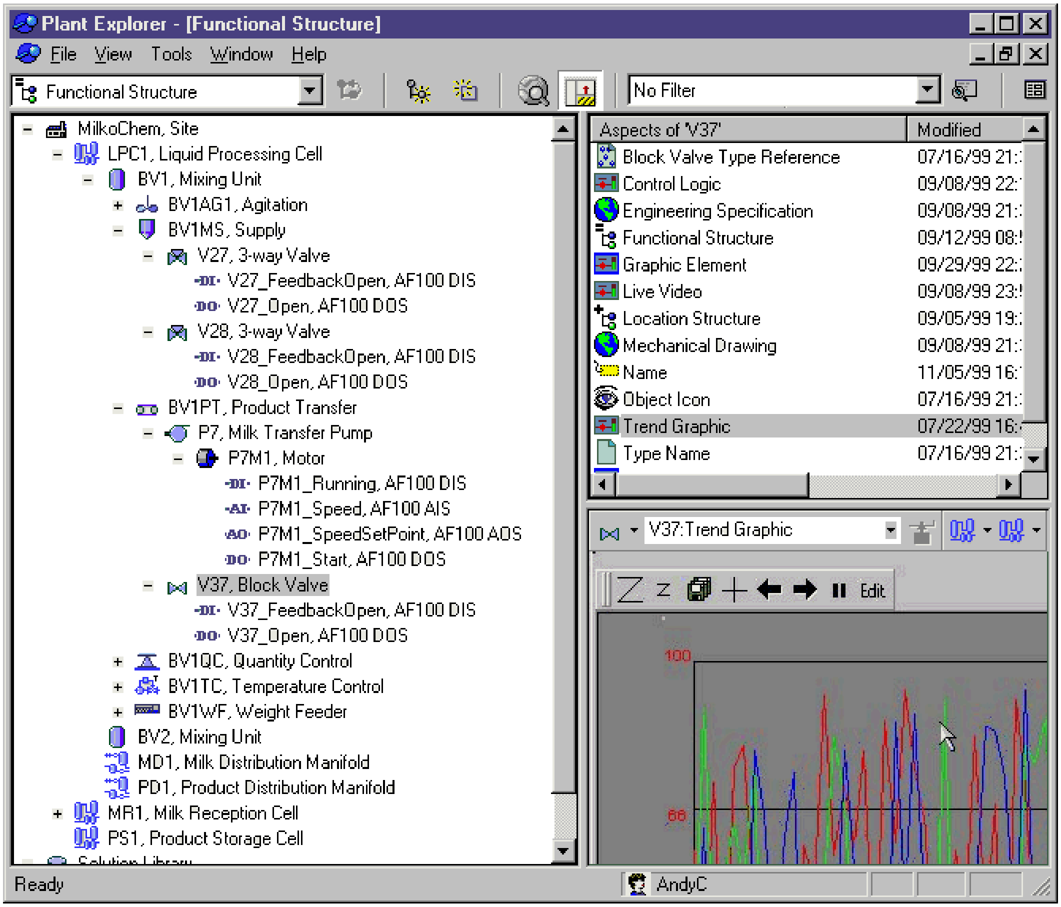
You assign application data to aspects of ABB Objects. The data of an aspect is handled by a corresponding aspect system. Edit aspect data by selecting the ABB Object and then clicking on the respective aspect.

Most of this work is described in:

- Chapter 9, Build the Location Structure
- Chapter 18, Building Alarm and Event Applications
- Chapter 19, Customizing User Environment, Advant Workplaces
- Chapter 20, Building History and Trend Applications
- Chapter 21, Building Complementary Applications.

8.2.1 Selecting an ABB Object

Select an ABB Object from the structure you have selected by single clicking on it. All its aspects are presented in the aspect list window in Advant Plant Explorer. In this case an aspect is selected in the aspect area and the preview area is opened.



- MilkoChem, Site
  - LPC1, Liquid Processing Cell
    - BV1, Mixing Unit
      - BV1AG1, Agitation
      - BV1MS, Supply
        - V27, 3-way Valve
          - V27_FeedbackOpen, AF100 DIS
          - V27_Open, AF100 DOS
        - V28, 3-way Valve
          - V28_FeedbackOpen, AF100 DIS
          - V28_Open, AF100 DOS
      - BV1PT, Product Transfer
        - P7, Milk Transfer Pump
          - P7M1, Motor
            - P7M1_Running, AF100 DIS
            - P7M1_Speed, AF100 AIS
            - P7M1_SpeedSetPoint, AF100 AOS
            - P7M1_Start, AF100 DOS
        - V37, Block Valve
          - V37_FeedbackOpen, AF100 DIS
          - V37_Open, AF100 DOS
      - BV1QC, Quantity Control
      - BV1TC, Temperature Control
      - BV1WF, Weight Feeder
    - BV2, Mixing Unit
    - MD1, Milk Distribution Manifold
    - PD1, Product Distribution Manifold
  - MR1, Milk Reception Cell
  - PS1, Product Storage Cell

Aspects of 'V37'	Modified
Block Valve Type Reference	07/16/99 21:
Control Logic	09/08/99 22:
Engineering Specification	09/08/99 21:
Functional Structure	09/12/99 08:
Graphic Element	09/29/99 22:
Live Video	09/08/99 23:
Location Structure	09/05/99 19:
Mechanical Drawing	09/08/99 21:
Name	11/05/99 16:
Object Icon	07/16/99 21:
Trend Graphic	07/22/99 16:
Type Name	07/16/99 21:

Figure 8-3. Selecting an ABB Object in Advant Plant Explorer

8.2.2 Creating New Aspects

8.2.2.1 Automatically Created Aspects

Figure 8-3 shows a number of aspects. Some of these aspects are created automatically when you create the object. Such aspects are:

- Name aspect
- Object Type Reference aspect
- The Structure aspect of the structure selected at creation time.

The Basic Properties Aspect is automatically created when an ABB Object is created. It holds a name for the object. Often other inherited aspects are present as well, for example an object icon.

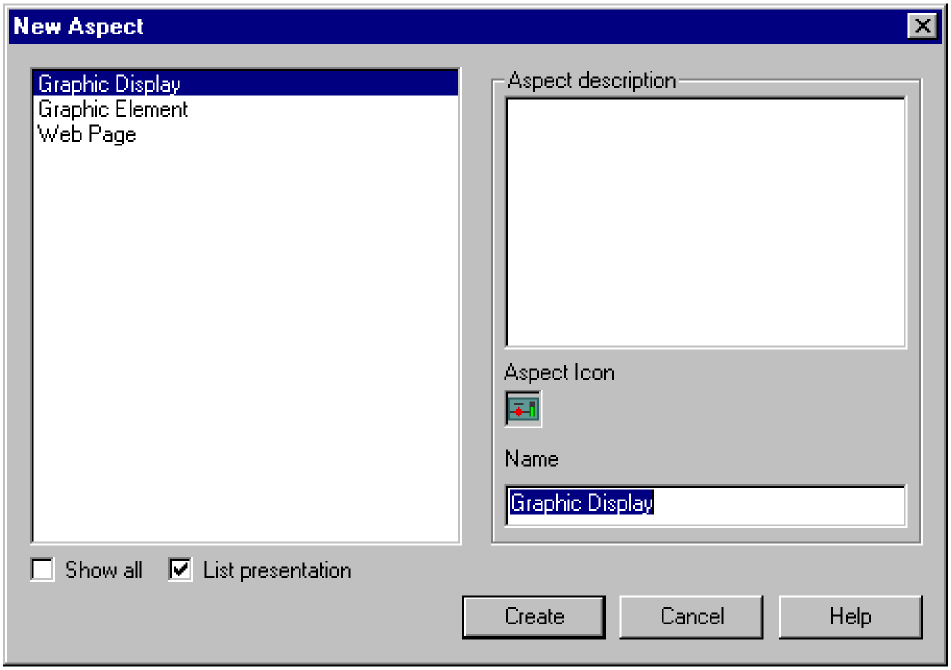
The structure aspects contains the information about the placement of an ABB Object in the existing structures. Opening a Structure aspect presents the ABB Object in the corresponding structure window, which of course is a very fast and elegant way to navigate.

8.2.2.2 Adding Aspects After Creation

Except for the aspects created by default the rest of the aspects are created later as you add more application data/functionality to the ABB Object. A large amount of pre-defined aspect categories exist. See the *Advant Plant Explorer User's Guide* where you can find an appendix with all the standard aspects in the release and the required user privileges to create, modify and view them.

To add an aspect to an ABB Object, do the following:

1. Make sure you have selected (clicked) the Object for which you want to add a new aspect.
2. Click the **New Aspect** button. A list of available aspect categories is displayed.



Name: Graphic Display

☐ Show all ☑ List presentation

Figure 8-4. New aspect dialog

3. Select a category.
4. If you like, you can also add a description for the aspect and a name for it. If you do not set a name, the aspect category name will be used as the aspect name.
5. Click **Create**. The aspect you have just created is presented as the topmost aspect in the list.

For more details and alternative ways of doing this, see the *Advant Plant Explorer User's Guide*.

8.2.3 Copying an Aspect

To copy an aspect, follow these steps:

1. Right-click the aspect you want to copy. The aspect context menu will appear.
2. Select **Copy**.
3. Right-click the Object to which you want to copy the aspect.
4. Select **Paste**.

(You can use drag and drop operations for Copy and Paste as well, see *Advant Plant Explorer User's Guide*).

8.2.4 Editing an Aspect

To gain access to the edit view of an aspect, double-click the aspect in the aspect list. The aspect view will be displayed in an overlapping window. If the preview option is selected, the aspects can also be edited in the Preview Area. To see the previews displayed, click the Preview Toggle Button on the tool bar. (For a layout overview, see Section 1.6.2, Layout of the Advant Plant Explorer Window.)

8.3 Adding Aspects According to Our Example

Continue with our example project. At this stage, there are not that many aspects to add yet. We can start documenting the plant though. The information used as basis for the creation and structuring of the upper two hierarchical levels in the function oriented structure, was taken out from the documents describing the plant and process. These documents should be assigned as aspects to the ABB Objects which are subject of the relevant document. Back to our example, have a look at the project object **Milko Chemical**.

The access to data describing the ABB Object is via the aspects of an object. Data that concern the project as a whole can be assigned to the **Milko Chemical** object. Translated to the ABB Object concept this means that you will add a number of aspects to the **Milko Chemical** object.

Such aspects are:

- Contract
- Scope of supply
- Description of the Designation System for the plant.

These are typically MS Word aspects. To handle them you create a Windows Application aspect, assign it to start MS Word with the respective document as parameter. (Requires that you have installed MS Office or at least MS Word on your PCs.)

The objects created in our example on the level below the Project object are specified by aspects representing Descriptions and Diagrams. These documents specify the process functions of the plant part represented by the objects and also the requirements for the electrical and control system.

Typical aspects on this level represent:

- Process diagrams (input to the graphic displays)
- P&I Diagrams
- Process Descriptions
- Functional Descriptions.

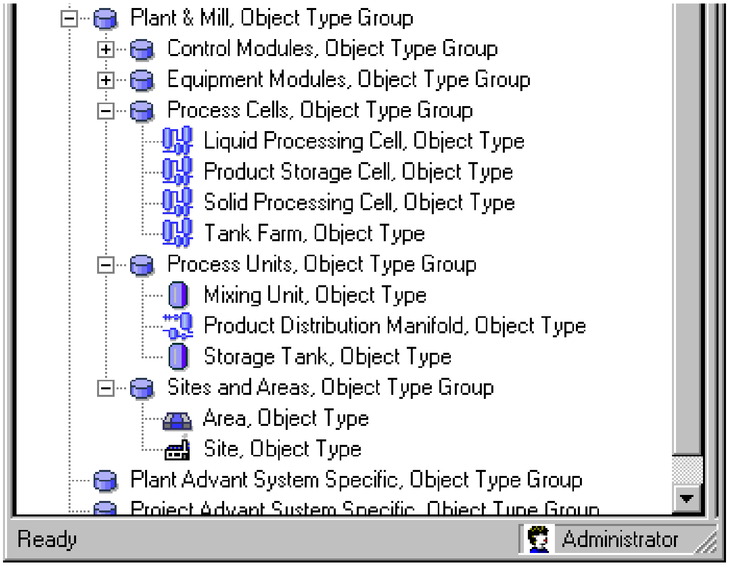
Once more you use the Windows application aspect to do this. If you need more details on how to use that aspect type, please read the *Advant Workplace User's Guide*.

You need to make sure that the Word documents are placed on a shared directory so that all the PCs in your configuration can access them.

8.4 Process Object Types to be Used in the Functional Structure

You can find the following object types useful for the building of the Functional structure. When you create a new object you can select show all, and you will get a structured list of object types to select from. In Section 8.4.1, Sites and Areas, Process Cells, Process Units, Section 8.4.2, Equipment modules and Section 8.4.3, Control Modules you can find the object types that exist in the standard product.

8.4.1 Sites and Areas, Process Cells, Process Units



Plant & Mill, Object Type Group
  Control Modules, Object Type Group
  Equipment Modules, Object Type Group
  Process Cells, Object Type Group
    Liquid Processing Cell, Object Type
    Product Storage Cell, Object Type
    Solid Processing Cell, Object Type
    Tank Farm, Object Type
  Process Units, Object Type Group
    Mixing Unit, Object Type
    Product Distribution Manifold, Object Type
    Storage Tank, Object Type
  Sites and Areas, Object Type Group
    Area, Object Type
    Site, Object Type
Plant Advant System Specific, Object Type Group
Project Advant System Specific, Object Type Group

Figure 8-5. Sites and Areas, Process Cells, Process Units

Here are the object types that are useful when you structure your plant.

- Site and Area - the top level
- Process cells - the next level - Liquid Processing Cell, Product Storage Cell, Solid Processing Cell, Tank farm

  Of course there are many other types of process cells but here are some examples. You can use these object types as input when you create new ones.

- Process Units - the next level again - Mixing Unit, Product Distribution manifold, Storage Tank

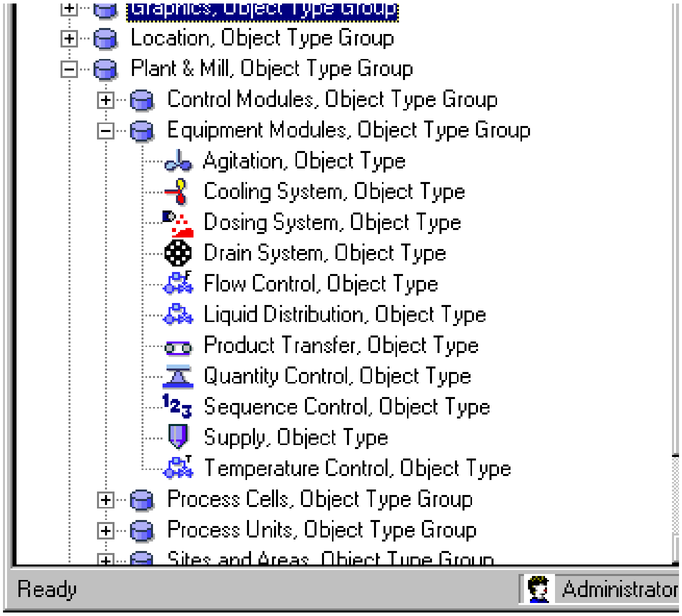
  As for the process cells there are many other types of process units. Here are some examples, you can use these object types as input when you create new ones.

8.4.2 Equipment modules

There are currently 11 object types represent equipment modules:

> Agitation, Cooling system, Dosing System, Drain System, Flow Control, Liquid Distribution, Product Transfer, Quantity Control, Sequence Control, Supply and Temperature Control.

See Figure 8-6.



Graphics, Object Type Group
Location, Object Type Group
Plant & Mill, Object Type Group
  Control Modules, Object Type Group
  Equipment Modules, Object Type Group
    Agitation, Object Type
    Cooling System, Object Type
    Dosing System, Object Type
    Drain System, Object Type
    Flow Control, Object Type
    Liquid Distribution, Object Type
    Product Transfer, Object Type
    Quantity Control, Object Type
    Sequence Control, Object Type
    Supply, Object Type
    Temperature Control, Object Type
  Process Cells, Object Type Group
  Process Units, Object Type Group
  Sites and Areas, Object Type Group

Ready | Administrator

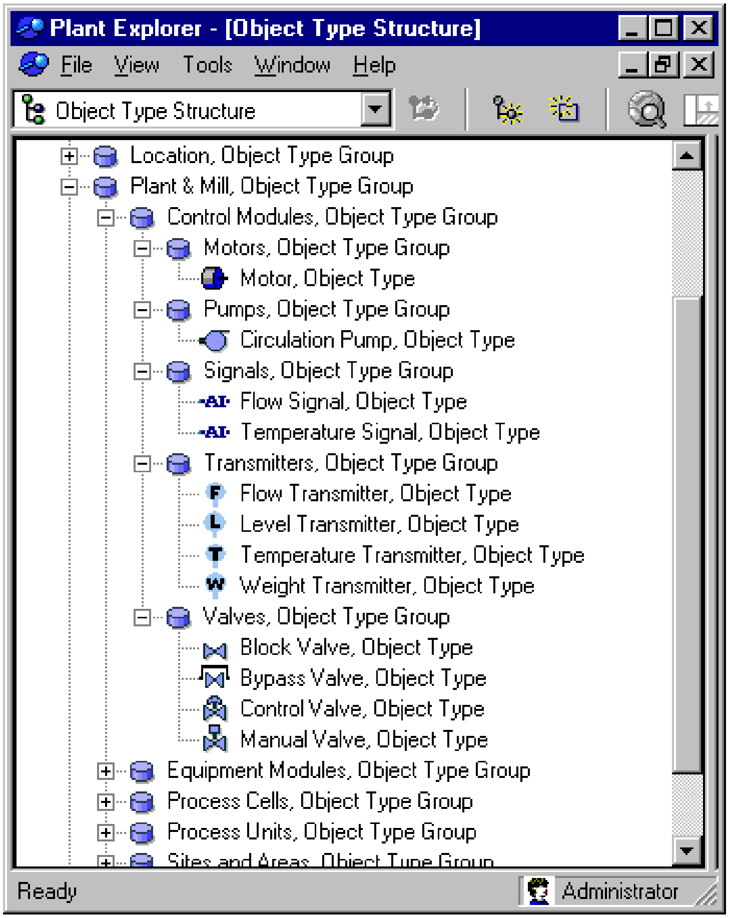
Figure 8-6. Equipment Modules

8.4.3 Control Modules

Object types are provided for the following Control modules:

- a motor group (one motor type)
- a Pump group (one pump)
- a Signal group (two signal types: Flow and Temperature)

- a Transmitter group: (four transmitter types: Flow, Level, Temperature and Weight)
- a Valve group (Block, Bypass, Control and Manual Valve)



- Location, Object Type Group
- Plant & Mill, Object Type Group
  - Control Modules, Object Type Group
    - Motors, Object Type Group
      - Motor, Object Type
    - Pumps, Object Type Group
      - Circulation Pump, Object Type
    - Signals, Object Type Group
      - Flow Signal, Object Type
      - Temperature Signal, Object Type
    - Transmitters, Object Type Group
      - Flow Transmitter, Object Type
      - Level Transmitter, Object Type
      - Temperature Transmitter, Object Type
      - Weight Transmitter, Object Type
    - Valves, Object Type Group
      - Block Valve, Object Type
      - Bypass Valve, Object Type
      - Control Valve, Object Type
      - Manual Valve, Object Type
  - Equipment Modules, Object Type Group
  - Process Cells, Object Type Group
  - Process Units, Object Type Group
  - Sites and Areas, Object Type Group

Figure 8-7. Control Modules

Chapter 9 Build the Location Structure

If your process/plant is distributed into several rooms, buildings or other location entities you should build up a Location structure to support navigation and location overview. If your process/plant is located in one large room/building and a location structure does not add any substantial information you can disregard this step. You can also wait until later to build the location structure.

There are two steps:

- Building up the overview Location structure.
- Complement the Location structure with existing ABB Objects. See Chapter 17, Complement the Location Structure.

9.1 Building up the Overview Location Structure

See Chapter 8, Build the Functional Structure. The steps to do are identical with the steps described for the Functional structure with one exception, there are some additional object types that you can use: Building and Room, see Figure 9-1.

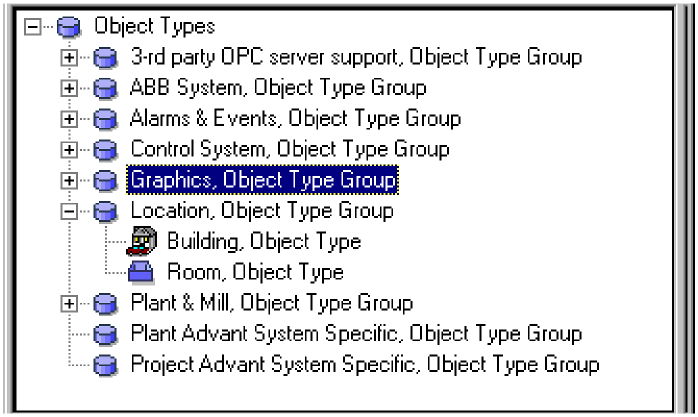

Object Types
3-rd party OPC server support, Object Type Group
ABB System, Object Type Group
Alarms & Events, Object Type Group
Control System, Object Type Group
Graphics, Object Type Group
Location, Object Type Group
Building, Object Type
Room, Object Type
Plant & Mill, Object Type Group
Plant Advant System Specific, Object Type Group
Project Advant System Specific, Object Type Group

Figure 9-1. Location object types

A good start is to use the default structure that is provided in the product - see Figure 9-2 - and then rearrange and rename it until the top levels with site and buildings are in place.

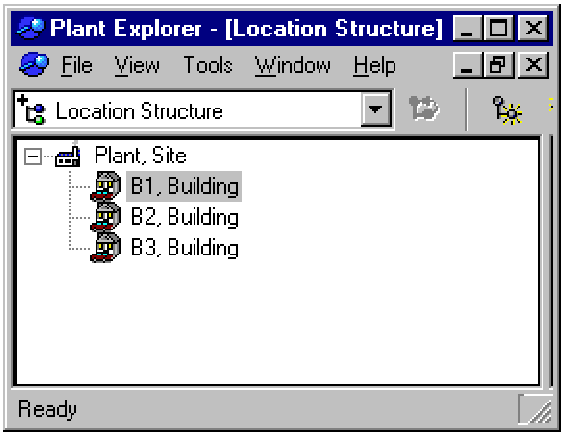


Figure 9-2. Default location structure

Then you can place the rooms in the respective buildings (if relevant) and you have the top structure. See Chapter 17, Complement the Location Structure for more how to continue.

Chapter 10 Build Graphic Displays

10.1 The Milko Milkshake Plant

In the previous chapters you have created the basic Functional structure and you have built whatever graphic elements, dialogs and typical solutions that you need to complement the optional libraries of Advant Graphics Builder. The building of graphic displays can be done step by step as more parts are to be included or in one step provided that all objects that it needs are available in the Functional and Control structure.

As a first step in the display building you need to decide which graphic displays that are needed for monitoring and control of this plant. Please see Appendix A, Example on Reference Material for details.

We decide that the following displays are needed:

- On the Milko Chemical object: **Milko Overview**
- On the LPC1 object: **Graphic Display** (describing the Liquid Processing Cell)
- On the BV1 object: **Graphic Display** (describing BV1)
- On the BV2 object: **Graphic Display** (describing BV2)
- On the PD1 object: **Detailed Graphics** (describing the Product Distribution Manifold)
- On the MD1 object: **Detailed Graphics** (describing the Milk Distribution Manifold)
- We already built the skeleton to a detail display over the Milk Reception Cell (MR1) in Chapter 5, Typical Solutions Overview.

10.2 Checking for Needed Building Blocks

Before you start building the Graphic Display you need to verify that you have installed the required software and/or have access to the necessary documentation:

- Visual Basic 6.0 is installed.
- The Advant Graphics Builder option is installed, probably (you normally need them) together with the following library options:
  - The Advant Graphic Library for Process Industry
  - The Advant Connectivity Library for AC100 Series
- You might have the following options installed:
  - Instrumentation ActiveX Library
  - Symbol Factory.

  For more information about these options see *Advant Workplace Products for Operators Product Guide*.
- You have most of the related objects described in the Functional structure.
- The typical solutions and object types that you need are built.
- The *Advant Graphics User's Guide.*
- A basic knowledge how to use Visual Basic[1] and Advant Graphics Builder.

10.3 Building the Milko Overview Graphic Display

Here follows a short description on what you need to do to build the Milko Overview Graphic Display.

The first thing you do is to decide upon a Process Object to place the Graphic Display aspect on. As you can see from the Appendix A, Example on Reference Material the obvious choice is Milko Chemical.

---

[1] To build ordinary displays you don´t need more knowledge of Visual Basic than how to use the “drawing board” and the drawing facilities.

The *Advant Graphics User's Guide* gives you all the necessary details on how to build, test and deploy displays. Here is a short overview for one of the displays:

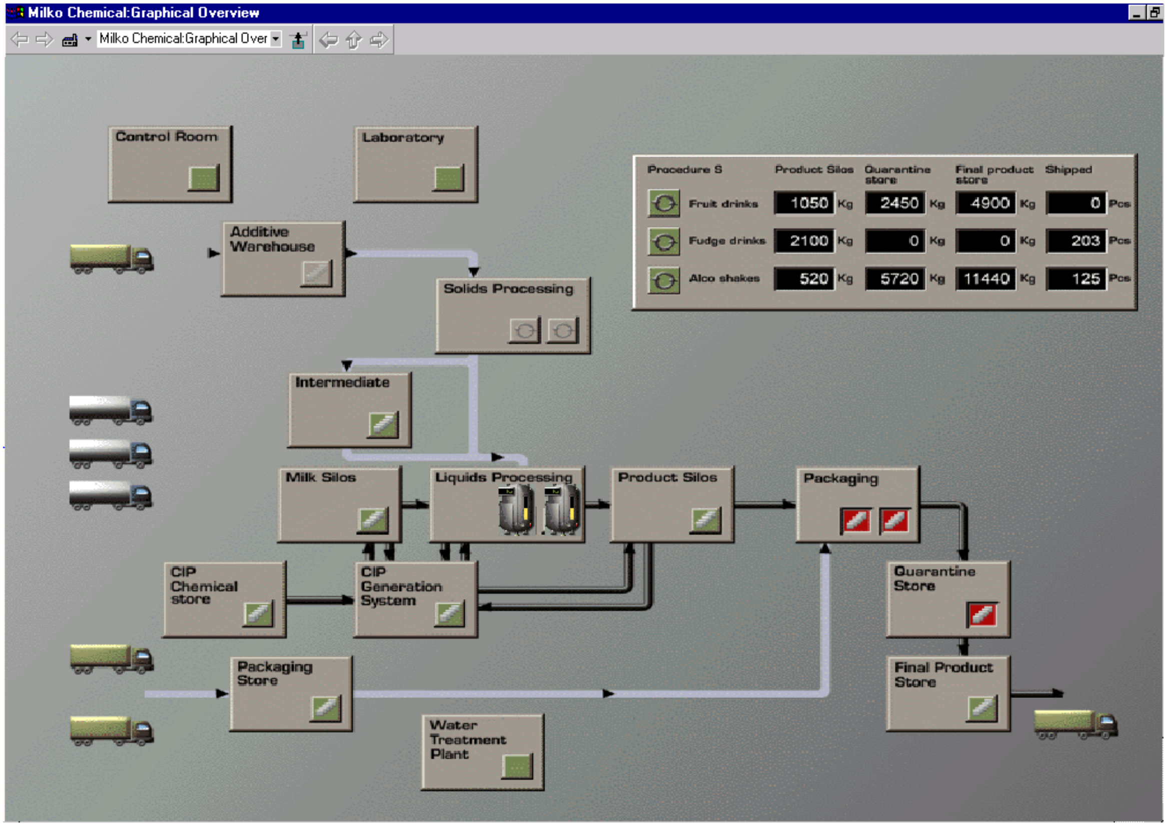
1. Choose Functional structure and the object Milko Chemical in Advant Plant Explorer.
2. Create a graphic display aspect.
3. Right-click the graphic display aspect and choose **Edit** from the context menu. This will start Advant Graphics Builder.
4. The first time you edit the aspect a wizard will help you to get the right settings to get started.
5. Advant Graphics Builder is now ready to be used. Select the tools you need from the toolbox and build it.
6. The first step is to build the background. Use Visual Basic together with the installed primitives and subelements, perhaps also the Symbol library.

   If you want to build your own animated graphics or import and filter photos, which are all useful additions to graphic elements/graphic displays, there are several third party products that can be used. We recommend Paint Shop Pro from JASC. Please look at http://www.jasc.com. There is a trial license for 30 days.
7. A very important function is the ability to quickly configure the presentation of the signal objects, not previously presented in the graphic display. The best way of doing it is to use the element browser:
   a. Click on the Add Element button. The new element is positioned at a default position.
   b. Click on the element and drag it to the spot where you want to place it.
   c. Select structure.
   d. Browse to the right object.
   e. Select the graphic element to use.
   f. Repeat steps a-e for all the elements you need to add.

8. As an alternative you can drag and drop using the Advant Plant Explorer:
    a. Start Advant Plant Explorer if you haven't started it before.
    b. Select the structure containing the objects you want to present.
    c. Browse to the object you want to present.
    d. Drag the preferred graphic element aspect into the graphic display in Advant Graphics Builder.
    e. Customize the presentation through the graphic element properties (An object aware element does not normally possess any properties that can be dynamically written to. Therefore, using expressions are of no use.).
    f. Repeat step c-e for all the signal object you want to add to the graphic display.

9. When you are done with all graphic elements, test the graphic display.
10. Save the graphic displays before you exit the application.
11. Deploy the graphic displays. Now it can be used on the other Advant Workplaces in your configuration.

When you are finished you will have results similar to Figure 10-1.



Procedure S	Product Silos	Quarantine store	Final product store	Shipped
Fruit drinks	1050 Kg	2450 Kg	4900 Kg	0 Pcs
Fudge drinks	2100 Kg	0 Kg	0 Kg	203 Pcs
Alco shakes	520 Kg	5720 Kg	11440 Kg	125 Pcs

Figure 10-1. Example of a Graphic Display

10.4 Building the First Parts of a Milk Reception Display

Here is an example of how to use the building blocks in Chapter 5, Typical Solutions Overview, to build the first part of the graphic display Milk Reception Cell.

1. Select the MR1 object and then press the New Aspect button in the toolbar.
2. Select the aspect category Graphic display and then press **Create**.

3. Right-click the Graphic Display aspect to get the context menu and select **Edit**, press **Finish** in the wizard that shows up.
4. Select **Tools > Element Browser**, select Functional structure expand down to Milko Chemical-MR1-MS1, select the Tank Element aspect of the MS1 Object and press **Add Element**.
5. Then select MS1DR1 and the Graphic Element aspect of the MS1DR1 Object and press **Add Element**.
6. Close the Element Browser.
7. Now place the tank element on top of the drain system element.
8. Repeat the steps 4-6 for MS2 and MS3.
9. Select **File > Deploy**.
10. Select menu item **File > Exit** to exit the Graphics Builder.
11. Double-click the graphics display aspect and show how the graphics appears.

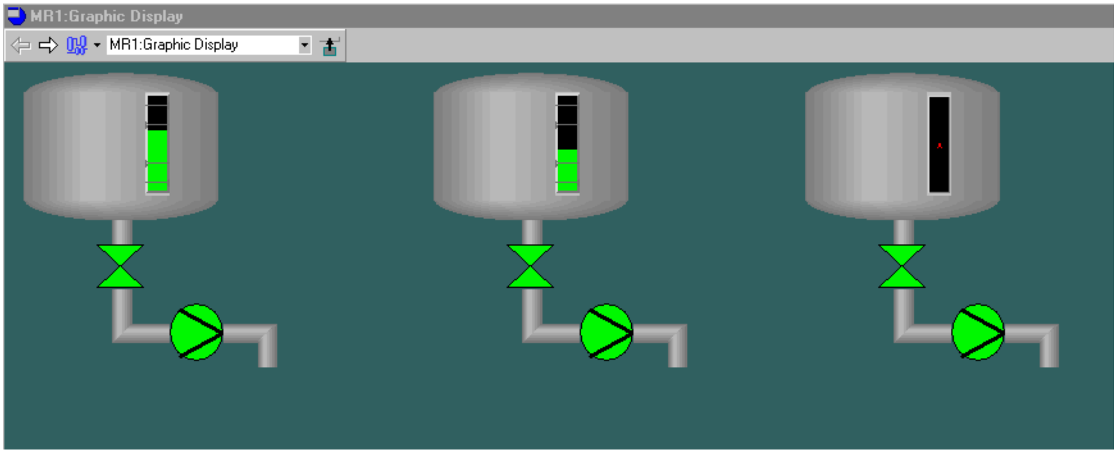


Figure 10-2. The MR1 Graphic Display after these steps

10.5 Building the Rest of the Displays

Repeat this procedure for each of the displays that you want to build until you have at least the skeleton for all displays.

When you build a display you should make sure to build it in the size that you normally intend to view it in. A small overlap should be built as such and a large overview graphic display should be built to cover the whole screen.

Chapter 11 Set up a Controller Configuration Project

11.1 Application Builder

Application Builder (APB) forms the base for controller configuration and lets you create, define and administrate your control configuration project.

From Application Builder you can start the other tools, Function Chart Builder (FCB) and Bus Configuration Builder (BCB), which are necessary to configure your system. For faster access you can also directly start the Builders from the corresponding folder of the Windows NT **Start** button.

You start Application Builder by selecting **Start > Advant > Engineering > Application Builder**.

11.2 Set up a Project in APB

Before you configure and program your controllers for your Milko project, you have to do some preparations.

You will have to create at least one project and for every Advant Controller, create a node by using the Application Builder. In one project, you can create several nodes.

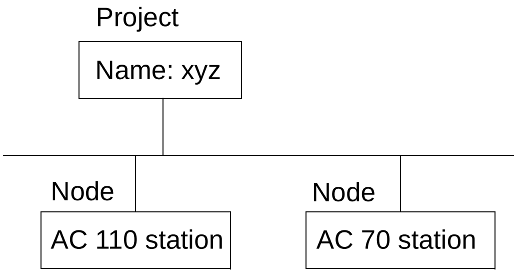
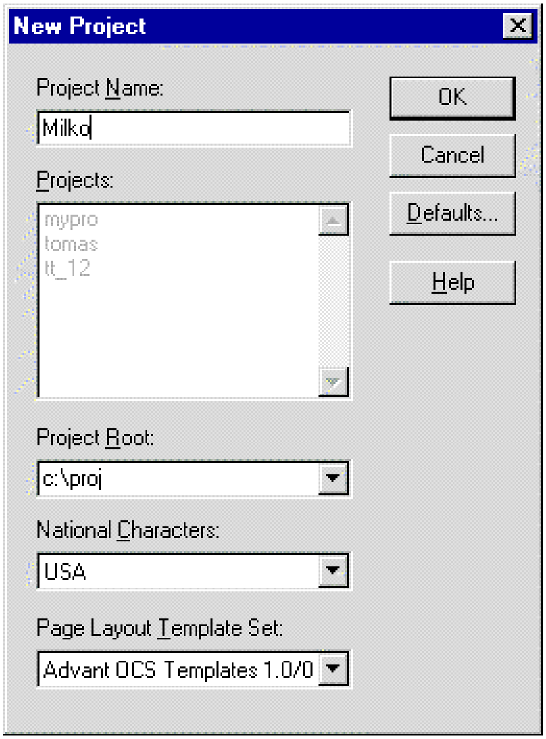


Figure 11-1. Project/Node structure on Advant Station 100 Series ES, Example

11.2.1 Create the Project

1. Use the **File > New > Project** menu command to generate a new project.

The following dialog is then displayed:



Project Name: Milko

Projects: mypro, tomas, tt_12

Project Root: c:\proj

National Characters: USA

Page Layout Template Set: Advant OCS Templates 1.0/0

Figure 11-2. Project Administration window

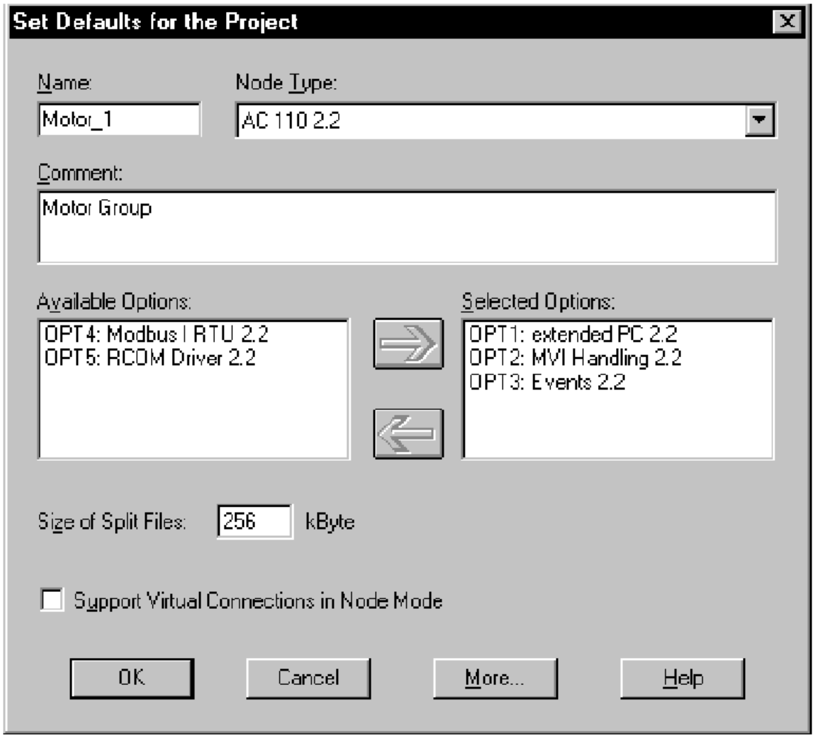
When a new project is generated an already open project is closed automatically, this is because working is allowed only with one project at the time.

2. Enter a project name, for example Milko. The maximum length is eight characters (a-z, A-Z, 0-9 and “_”).
3. All other properties can be left as they are, as defaults.

11.2.1.1 Project Defaults

Now it is time to set the project defaults. Click the Default button in the New project window or, in an existing project, use the **Options > Project Defaults** menu command.

The following dialog is then displayed:



Name: Motor_1

Node Type: AC 110 2.2

Comment: Motor Group

Available Options:
OPT4: Modbus I RTU 2.2
OPT5: RCOM Driver 2.2

Selected Options:
OPT1: extended PC 2.2
OPT2: MVI Handling 2.2
OPT3: Events 2.2

Size of Split Files: 256 kByte

☐ Support Virtual Connections in Node Mode

Figure 11-3. Project Default Window

All items in this dialog represent default values, which are presented in the according dialogs when you create new nodes, circuits and type circuits.

You can define the project defaults:

- One name for nodes/circuits/type circuits
- One node type (this includes also a possible change of the base software and the available options)
- One comment
- Selected options for *one* node type (Select the options you have license for).

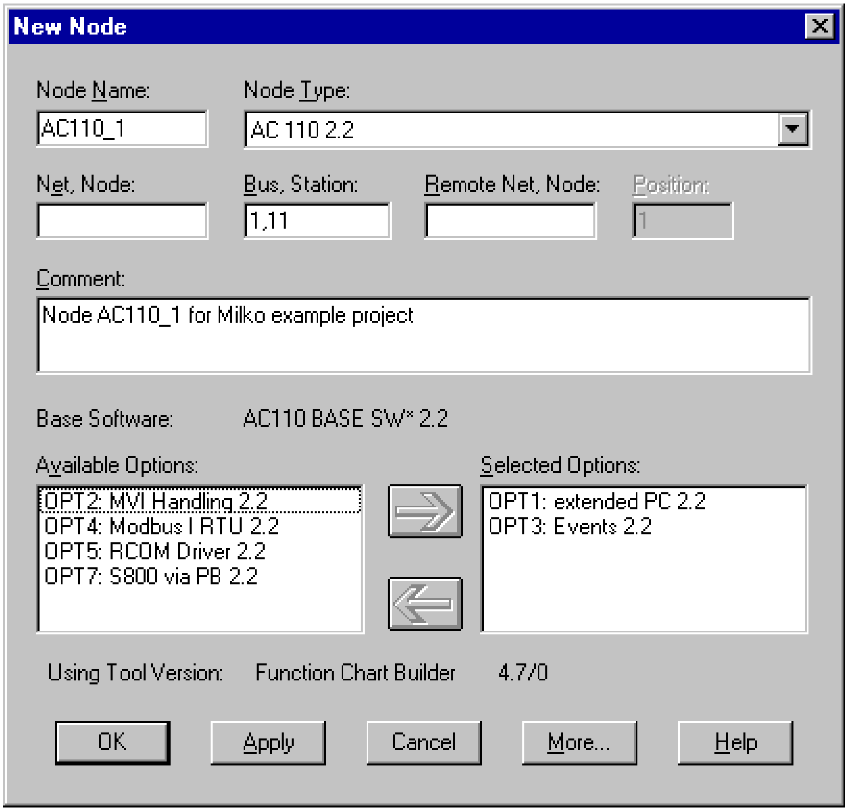
Further on you can define tool versions, document texts or page setup when you click the **More** button.

11.3 Create the Nodes in APB

When the project has been created you need to add the nodes.

1. If the project is not opened, open it.
2. Use the **File > New > Node** menu command to generate a new node.

The following dialog is then displayed:



New Node

Node Name: AC110_1

Node Type: AC 110 2.2

Net, Node:

Bus, Station: 1,11

Remote Net, Node:

Position: 1

Comment: Node AC110_1 for Milko example project

Base Software: AC110 BASE SW* 2.2

Available Options:
OPT2: MVI Handling 2.2
OPT4: Modbus I RTU 2.2
OPT5: RCOM Driver 2.2
OPT7: S800 via PB 2.2

Selected Options:
OPT1: extended PC 2.2
OPT3: Events 2.2

Using Tool Version: Function Chart Builder 4.7/0

Figure 11-4. New node Window

3. Enter AC110_1 as a name for the new node.
4. Choose the type of controller you have, for example AC 110 2.2, as node type.
5. Leave net, node address blank. In our Milko example, type 1,11 for bus, station and leave remote net, remote node, and position blank.
6. Select the options you have available on your controller, in our case it is OPT1 and the event handling option, OPT3.

In case of AC 110, 70, and derivatives, connected to an Advant Fieldbus 100 bus, the station number of the node is the controllers station number on the Advant Fieldbus 100 bus.
The station number is the number of the rotary (thumbwheel) switch in the basic station.

The setting for the bus number must be the same as the setting used for the Advant OPC Server for AF100 with which the controller is to communicate. The bus number for the Advant OPC Server for AF100 is normally set during the installation of the AF100 Interface.

7. Use the **Apply** button to leave the dialog open and save the changes. You have then the possibility to generate some more nodes.

After creating a new node the node list pops up automatically and the new node is inserted in the list.

You are now ready to start building your application program.

For more information about Application Builder, please refer to *Application Builder User's Guide*.

Chapter 12 Build the User Application Program

12.1 Overview

By using AMPL Control Configuration, which contains Function Chart Builder, and predefined functions in the controller, you produce the user application program.

The user application program for Advant Controller 100 Series consists of the following parts:

- Configuration part containing the DB elements of the related node
- Program part containing the PC elements of the related PC program.

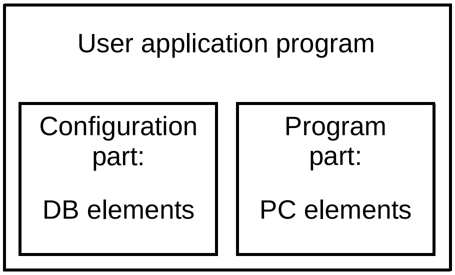


Figure 12-1. User application program for Advant Controller 100 Series

12.2 Starting the Function Chart Builder

To start Function Chart Builder, do as follows:

1. In the Application Builder, select the node, AC110_1, you created.
2. Start the Function Chart Builder by using the **Tools > Function Chart Builder** menu command.

   If you did not select any project default page layout in the Application Builder when creating the project, the Function Chart Builder presents the **Page Setup** dialog box, where you have to define the page layout for your diagrams.

3. Select the page layout for your diagrams (for example A4, landscape English):
4. Choose an appropriate page size, and click **OK**.

The Function Chart Builder presents the **Function Chart Builder** window showed below.

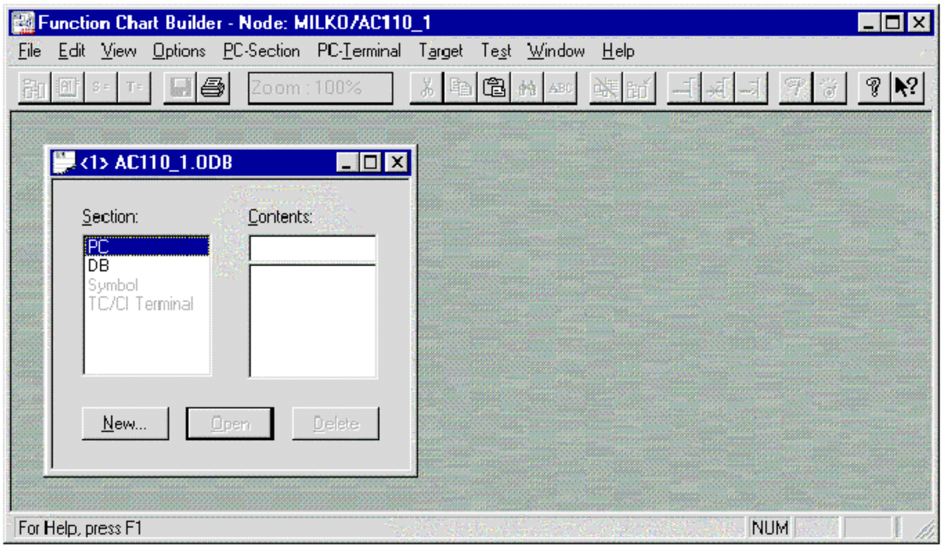


Figure 12-2. Function Chart Builder window

The **AC110_1.ODB** window lists the, now empty, sections of the node:

- PC section with generally none, one, or more PC programs consisting of PC elements.
- DB section generally containing DB elements.
- Node structures, circuits, and type circuits generally have a symbol section containing symbols used as parameters of type circuits and circuits.
- Circuits and type circuits generally have also a TC/CI terminal section containing symbols used as terminals of type circuits and circuits.

12.3 PC Program

The program part of a user application program describes the control algorithm and the control strategy. It contains the required PC elements, their interconnections, and the connections to the DB elements.

This part of the user application program is called the PC program. It consists of the PCPGM header element and subordinated structure and basic PC elements.

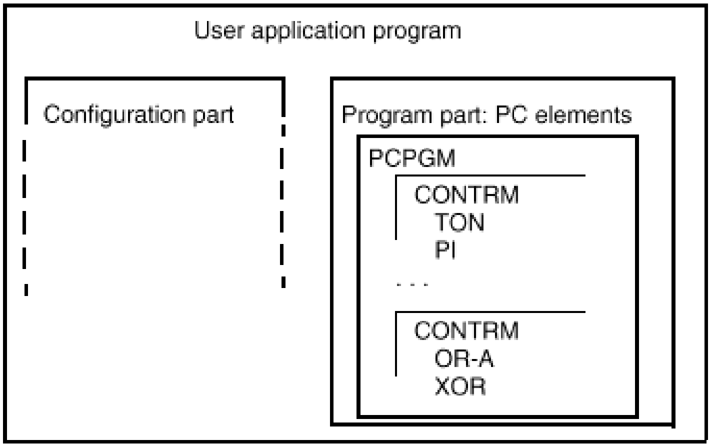


Figure 12-3. Program part within User application program

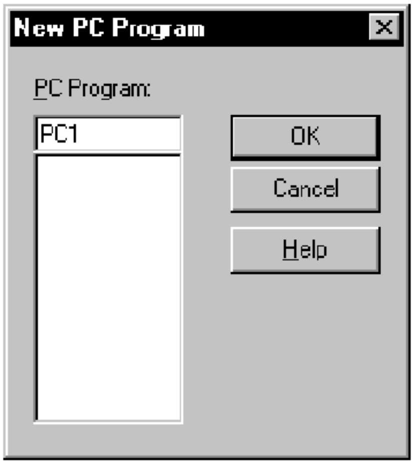
12.3.1 Create PC program and the Mandatory PC Elements

To start working on the PC program, first:

1. Create a PC program by selecting PC in the **Section** list box, Figure 12-2, and activating the **New** button. Alternatively, you can create a PC program by the **File > New Section > PC** command.

The Function Chart Builder displays the **New PC Program** dialog box.

2. Fill in the name in the **PC Program** text box, for example PC1, as shown in the following figure:



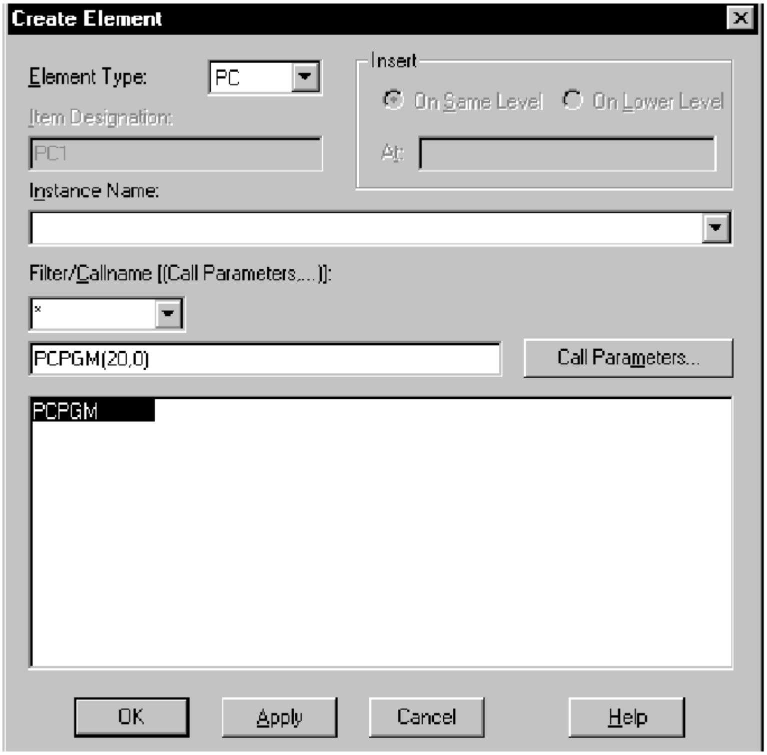
PC1

Figure 12-4. Defining PC1 as New PC Program

3. Click the **OK** button.

A number of mandatory PC elements must be created initially. A PC program (of a node or node structure) must always contain a PCPGM as top element. The Function Chart Builder displays the **Create Element** dialog box ready to create the PCPGM of PC1.

4. Fill in cycle time (20 milliseconds) in **Callname [(Call Parameters,…)]** as shown in the following figure:



Element Type: PC

Item Designation: PC1

● On Same Level ○ On Lower Level

At:

Instance Name:

Filter/Callname [(Call Parameters,...)]: *

PCPGM(20,0)

PCPGM

Figure 12-5. Create Element dialog box

5. Confirm with the **Apply** button.

   The **Create Element** dialog box now contains all PC elements that can be inserted on one level lower than the PCPGM.

6. Select CONTRM by clicking on it in the list box and activate the **Call Parameters** button.The Function Chart Builder displays the **Edit Call Parameters** dialog box.

7. Enter 300ms for the call parameter cycle time and confirm with the **OK** button.

You are now ready to build the rest of your PC program.

PC elements have input and output terminals to interconnect them and to connect them to DB elements or constants. Each element is represented in a PC diagram by a graphic symbol designed to describe the function of the element.

PC programming is performed as follows:

- Suitable PC elements are first selected from an element library and the corresponding symbols are entered in a draft diagram.
- The elements are then interconnected to provide the required function.

For more information see the *AMPL Configuration Reference Manual*.

12.4 Configuration by DB Elements

The configuration part, the DB part, of a user application program contains the configuration elements of the controller system:

- Hardware description
- Common data description
- Data exchange description.

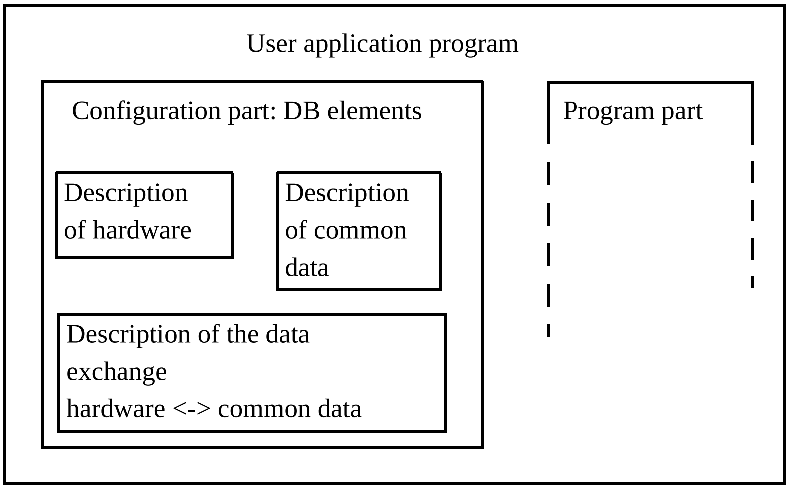

User application program
Configuration part: DB elements
Description of hardware
Description of common data
Description of the data exchange hardware <-> common data
Program part

Figure 12-6. Configuration part within User application program

12.4.1 DB Elements

DB elements in a controller system describe the following items:

- The hardware of the controller system:
  - CPU module, basic unit, expansion unit as applicable depending on the controller.
  - Digital input, digital output, analog input, analog output, and so on.
  - Communication interfaces, like MVI (Multi Vendor Interface) or connection to AF 100 as applicable for the controller.
- Common data elements:
  - DAT(B), DAT(I), DAT(IL), DAT(R), DAT(II)
  - PARDAT(B)
  - DAT_AI, DAT_AO, DAT_DAT, DAT_DI, and DAT_DO elements carrying HMI-relevant properties of signals.
- Event handling, EVS.
- Connection between the hardware and the common data elements as applicable depending on the controller: DataSet Peripheral DSP, MVI Data Block MVB.
- In case of AC 110 and derivatives also extended DB elements covering a set of the previous items and HMI functions: AIS, DIS, AOS, DOS, MB, MBS, MI, MIL, and MR.

12.4.2 Call Names of DB Elements

When you create DB elements you select them by their call names. Call names of DB elements describing modules and channels correspond to the identifier of the related modules (for example, AI610). For other elements, the call name equals the element name or is a short mnemonic of the element name.

Table 12-1. Examples of call names in Advant Controller 110

DB Element Type	Call Name(s)
Processor Module	PM632, PM633
Analog Input	AI610, AI620, AI625, AI630, AI635
Analog Output	AO610, AO650

For a complete list of DB elements see *Data Base Elements Advant Controller 70 Reference Manual*, *Data Base Elements Advant Controller 110 Reference Manual* and *Data Base Elements Advant Controller 160 Reference Manual.*

12.4.3 Create the DB Elements

Back to our example, Milko Chemical. We are no going to include the wanted DB element, by doing the following:

1. Create the DB section by selecting DB in the Section list box, Figure 12-2, and activating the New button. Alternatively, you can create a DB section with the **File > New Section > DB** command.

The Create Element dialog box is displayed.

2. In the list box, mark the desired database element e.g. PM633 (Processor Module for AC110).
3. Click the DB Terminal Values Button
4. Set terminal values, for example WITH_BAT to 1 if controller is equipped with battery backup, 0 if not.
5. Click **Apply**
6. Repeat to create the following DB elements:

   CI627 (Communication Interface for AF100) with the terminal values:

   BUSNO = 1 (in our example)

   STNNO = 11 (in our example)

   DI610

DO610

DIS with the terminal values:

NAME = V16_FeedbackOpen

SOURCE = HW

HW_REF = DI1.1

DS_COM = SEND

DSP_BUS = 1

DSP_STAT = 11

EV_COM = SEND, (Generate event sets and send them via EVS(S))

EVS_IDNT = 1

EVS_REF = 1

DSP_IDNT = 2, (Identity of DSP used)

DSP_CYCL = 512, (Wanted cycle time)

DSP_REF = 1, (Reference to the DAT element within the DSP)

If you are using AC 110, which is supporting extended DB elements for example DIS, EV_COM has to be set to SEND. This setting will automatically create, distribute and send EVS(S) to the OPC Server.

If you are using AC 70 you have to create the EVS(S) DB elements yourself. The DAT_DI, DAT_DO and for the alarm a DIC element together with an EVS, DSPs and DATs elements have to be used. See *Data Base Elements Advant Controller 70 Reference Manual*.

MMI_USE = YES

DESCR = Valve16isOpen

ERR_TR = 2, (0=no event, 1=event no alarm, 2=event and alarm)

VALUE_TR = 1, (0=no event, 1=event no alarm, 2=event and alarm)

For more information on event generation see Chapter 18.1

NORM_POS = 0, (Event generated when value differs from value stated here)

DOS with the terminal values:

NAME = V16_Open

SOURCE = MMI

HW_REF = DO1.1

MMI_USE = YES

DESCR = OpenValve16

7. To define the terminal values, click the **DB Terminal Value** button.

The Edit DB Terminal Values dialog is displayed. There can exist several parts of the DB element.

8. Choose the different parts from the drop list.
9. Fill in the desired terminal values.
10. Acknowledge with the OK button.

For more information on Event DB elements see Section 12.5, More Information on DB Elements and *AMPL Configuration Advant Controller 100 Series Reference Manual*.

If you are working with an AC 70 or AC 160 you have to select other PM and CI elements.

AC 70 and AC 110 and their derivatives provide event detection and distribution of time-tagged events over AF 100.

If you have the need for additional DB elements, add and configure them as well. You are now ready to generate and download your target code.

12.5 More Information on DB Elements

If you want to know more about DB Elements, keep on reading this chapter. If you want to continue with the Milko project and download your controller configuration application jump to chapter 13.

12.5.1 Event Detection

Kinds of Time-tagged Events

Advant Controller 100 Series controllers generate the following time-tagged events:

> "Calculated" events which generation is configured by DB elements AIC and DIC.

In case of AC 110 and derivatives, there are events generated also by other sources:

- Events from I/O modules, such as DI650/DI651/DI652
- Events received in MVI Data Blocks and converted into the internal event format by means of DB elements AIMVB and DIMVB
- Events from extended DB elements, such as AIS, DIS.

DB elements DIC, AIC, DIS, AIS, DIS650/DIS651/DIS652, DIMVB, AIMVB used for configuration of event detection functions are called **Event Channel** elements.

"Calculated" Events

DB elements DIC and AIC allow to watch connected PC signals of type Boolean or Real for events. Event detection is "calculated" according to the parameters specified at these DB elements with a resolution down to 10 ms.

An event is detected from a Boolean PC signal when the signal changes from 0 to 1 or vice versa.

From a PC signal of type Real an event is detected when the signal exceeds one of the two low or high limits or moves back into the range.

Event Queues

Events detected on controller are buffered in up to 32 event queues, each event queue can store up to 1000 events, depending on controller type. If events are generated while the event queue is full the new events overwrite the oldest events.

The event queue is represented by the EVS DB element automatically generated by AIS or DIS.

12.5.2 Event Transfer

General Features

Depending on the type of controller, time-tagged events from a controller are transferred to one or several event receivers over Advant Fieldbus 100. An event receiver can be:

- An Advant Controller
- A personal computer running a Human Machine Interface, for example, AdvaSoft 2.0 for Windows NT.

The number of event receivers on an AF100 is limited to 10.

A possible network configuration for event transfer is shown in Figure 12-7:

Event Set Elements

The transfer of time-tagged events is configured by Event Set (EVS) elements.

An Event Set element of type “send” (configured by data base element EVS(S)) groups a set of Event Channels (AIC, DIC, and so on, as applicable) for transmission of events. Each Event Set can handle up to 32 Event Channels of different types which can be mixed in arbitrary order.

An EVS(S) element collects events from its referenced Event Channel elements and sends the events to the event receivers when those request them.

An AC 110 or derivative can handle up to 32 Event Set elements EVS(S). For DB elements EVS(S) the Event Channel elements (AIC, DIC, AIS, DIS, DIS650/DIS651/DIS652, AIMVB, DIMVB as applicable) are considered as event sources. An AC 70 or derivative can handle up to 32 Event Set elements EVS(S).

Event Transfer to HMI

AC 110, 70, and derivatives transfer events directly to a personal computer running AdvaSoft 2.0 for Windows NT via AF 100.

12.5.3 Configuration of Event Sets

In Figure 12-7 a configuration example is shown for event transfer from an Advant Controller 100 Series controller and an Advant Controller 70 Series controller to a PC running AdvaSoft 2.0 for Windows NT, using AF 100.

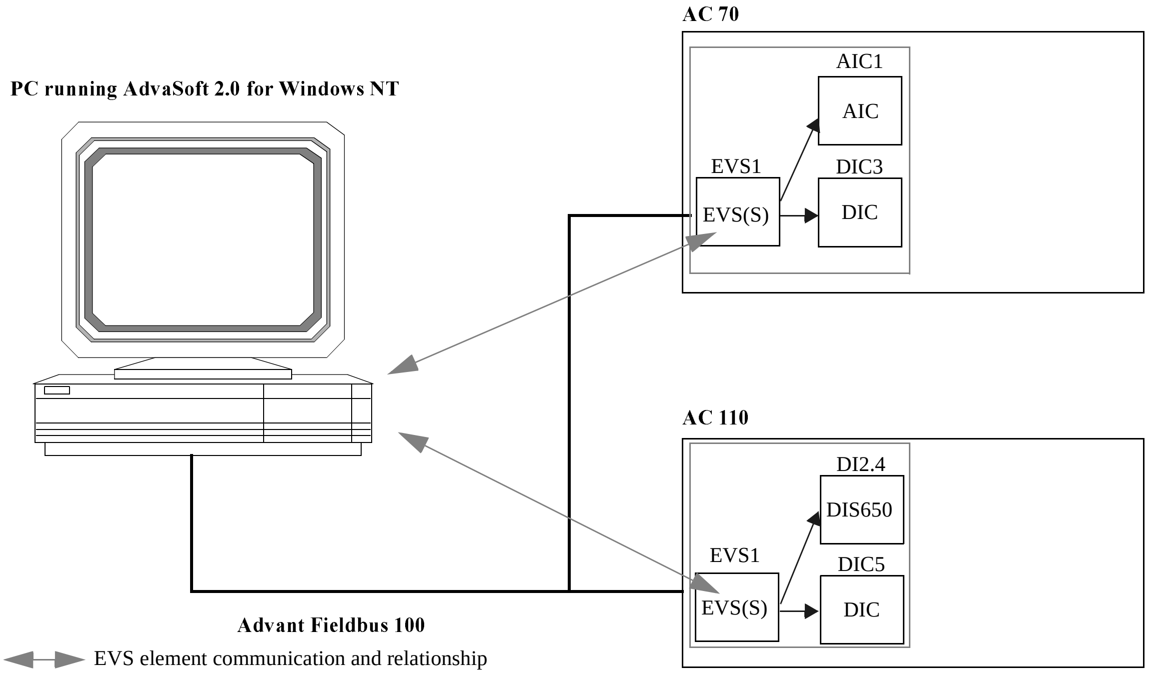


Figure 12-7. Event Transfer from AC 110, 70 to PC with Advant Operator Workplace (example)

For a description of how to configure the AIMVB, DIMVB, and DIS650/DIS651/DIS652 elements, see *AMPL Configuration Advant Controller 100 Series Reference Manual*.

12.5.4 Configuration of Extended DB Elements

12.5.4.1 Overview

Extended DB elements are used to simplify the communication between the controller and AdvaSoft 2.0 for Windows NT as well as the communication between different controllers.

The extended DB element is a concept that facilitates the engineering of AC 110 or derivatives. It addresses the following issues:

- Definition of signals independently of the I/O hardware.
- Definition of signals and texts for HMI (MMI) purposes. Every aspect of the signal, from DSP communication up to event generation can be seen at a single DB element.
- Distribution of signals on the AF 100. There is only one point in the system where the signal is defined. The definition includes also the conversion of raw values to engineering units.

The intention behind extended DB elements is the single point data entry in order to configure the controller as well as the HMI.

Extended DB elements carry information concerning the following items:

- Controller configuration
- HMI configuration
- Communication
- Signal description.

The following extended DB elements are available for AC 110:

- Input elements
  - Analog input (AIS)
  - Digital input (DIS).
- Output elements
  - Analog output (AOS)
  - Digital output (DOS).

- Memory elements (MB, MI, MIL, MR).

To work with extended DB elements with all their advantages, you need the following three systems:

- Controller with base software or options supporting extended DB elements, such as AC 110 (version 2.1 or higher) or AC 160
- AdvaSoft 2.0 for Windows NT together with Advant OPC Server for AF100
- Engineering environment supporting extended DB elements, such as Function Chart Builder version 4.3/0 or higher.

If you create, modify, and delete extended DB elements, FCB automatically creates, modifies, and deletes other DB elements, as required by the extended DB element.

Configuration of extended DB elements is done in two ways, depending on whether the configuration parameter is changeable at run-time by HMI:

- Some terminals have fixed values that can only be changed by FCB.
- Some terminal values can be changed by HMI.
  For example, HMI can initialize them at run-time by writing initial values.

We describe the usage of extended DB elements from the following points of view:

- Selection of signal sources, connection to signal sources
- Internal processing in extended DB elements
- Connection to signal sinks

Figure 12-8 shows the main system components participating in the realization of extended DB element concept.

12.5.4.2 Extended DB Element Communication Between Stations

Extended DB elements are very versatile because:

- Extended DB elements allow transfer of I/O signals or signals generated by PC programs (for example, output by PC elements) to HMI system and AC 110 (or derivative).
- They also allow transfer of commands and calculated data from HMI system to AC 110 (or derivative).

- In case of AC 160, extended DB elements also support data exchange between redundant CPUs within AC 160 stations via MDAT elements.

The following figure shows how the AIS extended DB element can be used to access and propagate an analog signal in different AC 110 (or derivative) stations:

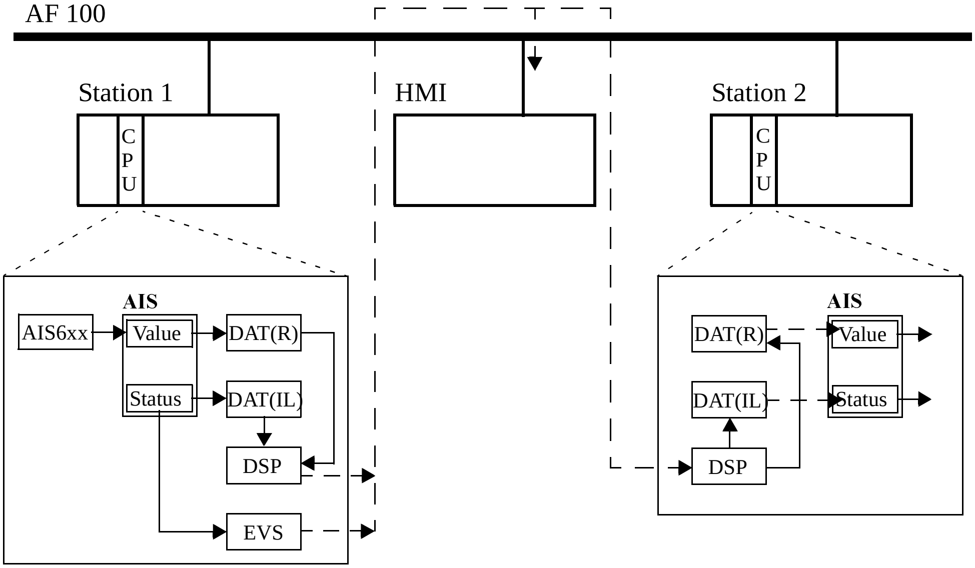


Figure 12-8. Extended DB element referencing HW element

The AIS in Station 1 is configured to fetch the value from a hardware input channel and send it via DS/DAT communication. Here, the extended DB element is referenced also by an EVS(S) element that provides the event queue. DSP, DAT(R), DAT(IL), and EVS are created automatically at the creation of AIS. AIS6xx, for example, AIS630, has to be created explicitly.

The AIS in Station 2 is configured to receive the value from Station 1 via DS/DAT communication. DSP, DAT(R), and DAT(IL) are created automatically at the creation of AIS.

12.5.4.3 Event Handling with Extended DB Elements

To allow an extended DB element to generate events, set its EV_COM terminal to SEND. Fill also the EVS_IDNT and EVS_REF terminals appropriately.

Then FCB creates the necessary EVS(S) element automatically.

The extended DB elements can generate events in the following cases:

- Operational changes
- Error conditions.

Examples

1. The following events are generated by an **AIS** element:
   - Limit violation (HI2, HI1, LO1, LO2)
   - AIS6xx error
   - Alarm blocking
   - Forcing
   - Acknowledge.
2. The following events are generated by a **DOS** element:
   - DOS6xx error
   - Forcing
   - Acknowledge
   - Value change.
3. The following event is generated by an **MB** element:
   - Value change (any of the 32 bit)
   - Acknowledge.

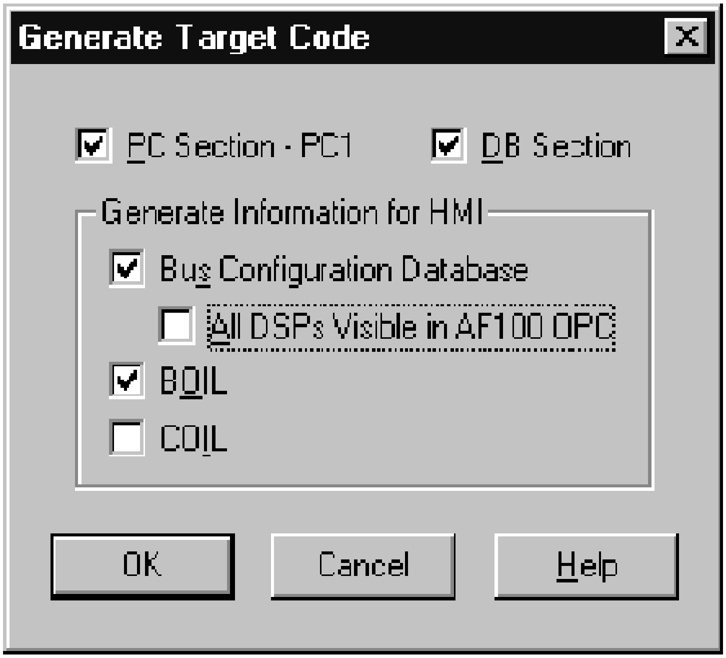
For more information on DB elements in general see *Data Base Elements Advant Controller 70 Reference Manual*, *Data Base Elements Advant Controller 110 Reference Manual* and *Data Base Elements Advant Controller 160 Reference Manual*.

Chapter 13 Generate and Download Target Code

13.1 Generate Target Code

After your user application program is ready, the target code of the user application must be generated and then downloaded to the controller station.

1. Generate target code for Advant Controller 100 Series, from the FCB´s internal data representation (ODB format), by the FCB command **File > Generate Target Code**.


Generate Target Code

☑ PC Section - PC1 ☑ DB Section

Generate Information for HMI
- ☑ Bus Configuration Database
  - ☐ All DSPs Visible in AF100 OPC
- ☑ BOIL
- ☐ COIL

Figure 13-1. Generate Target Code dialog box

2. Check the same four boxes as above in the Generate Target Code dialog box. Here it is especially important that the Bus Configuration Database box is checked. This command generates a file used by the Advant OPC Server for AF100. See Chapter 14 Restart the OPC Server for information on what to do with the file generated.
3. Click **OK**.

Besides generating target code, this function generates the contents of the Bus Configuration Database. For most node types, because of the controller memory restrictions, information concerning symbolic names, UNITS and so on, is generally not stored within the target code.

For information about APB and FCB, see *Application Builder User's Guide* and *Function Chart Builder User's Guide*.

13.2 Backup and Restore

13.2.1 Application Program in Source Code Form

For back-up reasons it is a good idea to generate and store the user application program for the controller in the engineering station or on a backup media.

You can store a user application program for the controller in two forms:

- Internal data representation of the Function Chart Builder (ODB format), which can be read generally only with FCB of the same release and revision as the one that stored it.
- AMPL source code (ASCII data file), compatible between FCB releases.

By using the FCB command **File > Generate Source**, you can generate AMPL source code from the FCB´s internal data representation.

By using the FCB command **File > Backtranslate Source**, you can convert the AMPL source code into the FCB´s internal data representation.

If you lose your data on the engineering station and there is no backup, you can neither document the actual program, nor make on-line changes. You must re-create your entire program on the engineering station.

13.2.2 Save and Backup

It is strongly recommended to save the actual user data of the engineering station on a backup medium, for example, floppy disks or network drives.

It is a good practice to do the following:

- Save the user data frequently during the creation of the application program.
- Save the user data when the creation of the application program on the engineering station is finished.
- Save the user data after every change within a “finished” application program.

To save and back up:

1. Use Application Builder commands **File > Copy Project** to save the whole project.
2. Then use the FCB command **File > Save as** to save the node you are working on.
3. In some cases a packing utility can be good to use.
4. Store the backup medium in a safe place.

13.2.3 Restore

To restore, use the Application Builder commands **File > Copy Project**, in some cases it can be good to use some unpacking utility on the engineering station.

13.3 Target Control Functions

On the engineering station, the Function Chart Builder offers commands in the **Target** menu for the following functions:

- Execution control of the PC program
- Maintenance features of the controller
- Maintenance features of the application.

Loading the application with the command **Target > Load Application** is the first step toward establishing an application on the Advant Controller 100 Series target.

Start the execution of the application with the command **Target > Deblock Program**.

Stop the execution with the command **Target > Block Program**.

Trigger a cold start with the command **Target > Restart**.

Back up the application on a PROM with the command **Target > Save in PROM**. Delete the contents of the PROM with **Target > Clear PROM**.

The **Target > Set Date and Time** command should not be used from here as you are using AdvaSoft 2.0 for Windows NT, which has its own functions for this. See the Advant Workplace Products Administrator’s Guide for more information

13.3.1 Load Application

To load the application, do the following:

1. Choose the **Connect** item from the **Target** menu in the Function Chart Builder.
2. A dialog is displayed with various connection alternatives. In the case of a totally new AF100 Interface installation and a totally new controller, use a serial communication line. Otherwise use the **Assistance** button to select the way you want to select.
3. Select the appropriate COMx alternative (depending on which port you are using) and click **OK**. You can use the AF100 alternative in all future application program download operations. After connection you get a report about the status of the connected controller in the Status Report window.

When updating an application that has previously been loaded in the controller you must first BLOCK PROGRAM.

4. The command **Target > Load Application** downloads a complete user application program to the connected AC 100 Series target.
5. After the download is finished, the status of the target is updated.
6. If the download is successful, the Function Chart Builder offers you to switch to on-line mode. See Section AMPL Configuration Advant Controller 100 Series Reference Manual, About On-line and Off-line Modes and Mode Transitions.
7. Select **Target > Save in Prom** to save the application program in the prom of the controller, as a backup
8. After you are done, choose the **Debacle Program** item from the **Target** menu.
9. Exit the Function Chart Builder application by choosing **File > Exit**.
10. A dialog is shown which asks if you want to disconnect from the target system. Enter Y for Yes.

Now you have all preconditions on the controller and in Bus Configuration Database fulfilled, to see "feedbackopen" and to control "open" on your AdvanSoft 2.0 for Windows NT. See Chapter 14 Restart the OPC Server for information on what to do with the Bus Configuration Database generated.

An off-line generated user application program is downloaded to the target system as one unit. There is no need to download the DB and PC elements separately.

Chapter 14 Restart the OPC Server

14.1 General

The Advant OPC Server for Advant Controller 100 Series does not support on-line additions of information. It uses the so called BCD.MDB (Bus Configuration Database) file as a source for its knowledge about the Controller objects. When the OPC server is started it reads the BCD.MDB file and builds up a list of signal objects and their attributes. This information is used by the AdvaSoft 2.0 for Windows NT software to populate the Control structure.

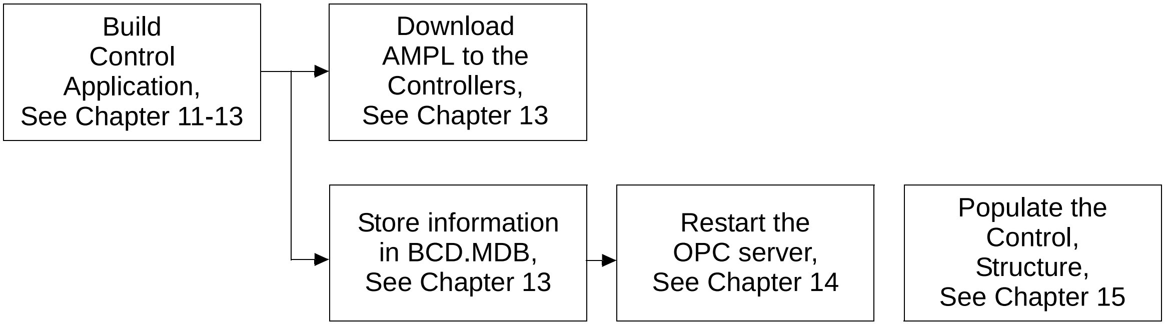

Build Control Application, See Chapter 11-13
Download AMPL to the Controllers, See Chapter 13
Store information in BCD.MDB, See Chapter 13
Restart the OPC server, See Chapter 14
Populate the Control, Structure, See Chapter 15

Figure 14-1. Creating the Controller applications

To make use of the updated information in the Controller, the OPC server needs to be restarted and the Control structure needs to be populated, see Figure 14-1.

To restart the OPC server is a step that will not exist in future releases. It is a limitation in the current release of the Advant OPC server for Advant Fieldbus 100 that will be removed a s a p. This also means that the storage into the MDB file will rather be a request from the engineering tool to update the OPC server information with the changes that have been done and with a notification when the Controllers are downloaded with the new information.

14.2 Restart the OPC Server

As a restart of the OPC server prevents monitoring and operation of a plant during the restart time you have to be careful when you do this. Especially if you are working in a multinode configuration.

To restart the OPC server do as follows:

1. Make sure that you have Administrator access rights to the PC.
2. Start the Task Manager and find the process called “wiaopc.exe” and select only that process.
3. Open the Control Panel by selecting **Start > Settings > Control Panel > Services.**
4. Find the service called “Advant OPC Server for AF100”. Make sure it has status **Started** and then press the **Stop** button.
5. Change quickly back to the Task Manager and verify that the process “wiaopc.exe” decreases its memory consumption and finally disappears from the process list.
6. Now, you can change back to the Control Panel and the Services dialog and press the **Start** button for “Advant OPC Server for AF100”.

The Service Manager might launch a message box claiming that the service could not be stopped. This message can be disregarded if the Task Manager behaved correctly in step 5 above. The reason is simply that the Service Manager expects the service to terminate within a certain amount of time (about 20 seconds). The “wiaopc.exe” process might need longer time to disconnect from all controllers and free all its allocated memory. In large configurations you will most likely get this message all the time.

If the procedure above gives you any problem, the entire system should be restarted for safety reasons.

Chapter 15 Build the Control Structure

15.1 Populating the Control System Structure

The Control structure describes the structure of the Control network, it´s Controllers and signal objects. The Control structure is used also to describe how communication with the Controllers (signal objects) shall be done by creating the corresponding Network objects. This is described in the Advant Workplace Products Administrators Guide where you can also read about the possibility to use an AF100 Test OPC Server if you do not have a real network connected yet. The Control structure also contains the PCA aspect on each signal object which is a must for a functioning communication.

In future releases the Control structure is built up step by step as the engineering tools are used to specify nodes and signal objects. In this release however this information has to be imported from the information stored by the engineering tools in the so called BCD.MDB file. See Chapter 13, Generate and Download Target Code for a description of what to do when the target code and the MDB.BCD file is created/updated.

As a middle step you also need to restart the OPC server, see Chapter 14, Restart the OPC Server. In future releases the changes in the controller will be detected as the Control engineering tools do them and as soon as the new target code is downloaded to the controllers new objects and their attributes will be added automatically.

The AF100 OPC Server *must* be running at this time. When you upload object, all objects that are created in this release are of the object types presented in Figure 15-5.

Steps to upload objects are the following:

1. Select **Start > Programs > Advant > Advant Plant Explorer**.
2. Select **Control structure** from the Structure selector.
3. Create an object of the AF100 Network Object type.

4. Select the **AF100 Network Object**, by the name you entered above, then select the **Uploader** aspect.
5. Click **Start Upload** on the Standard Tab.
6. Save changes by clicking **Apply**. Close the Aspect by clicking the [X] of the aspect window.

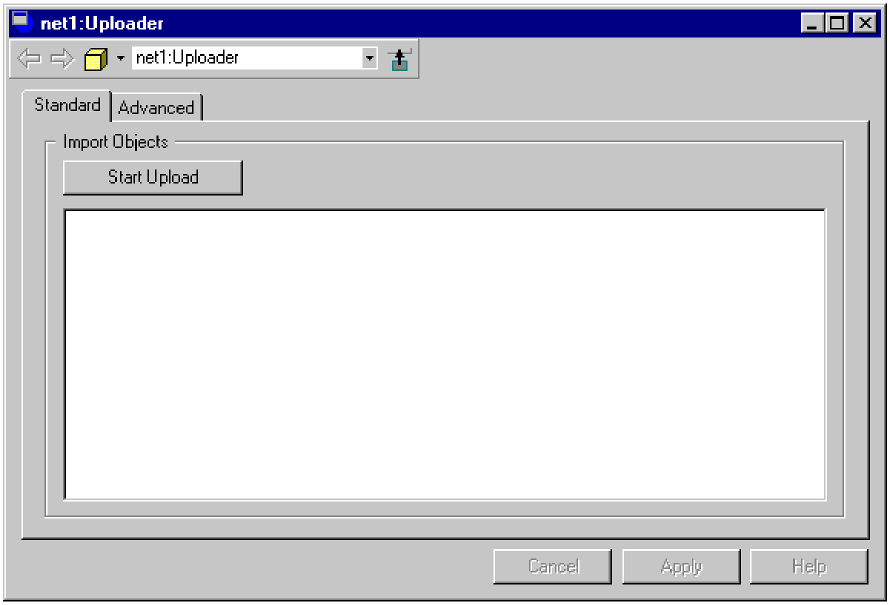
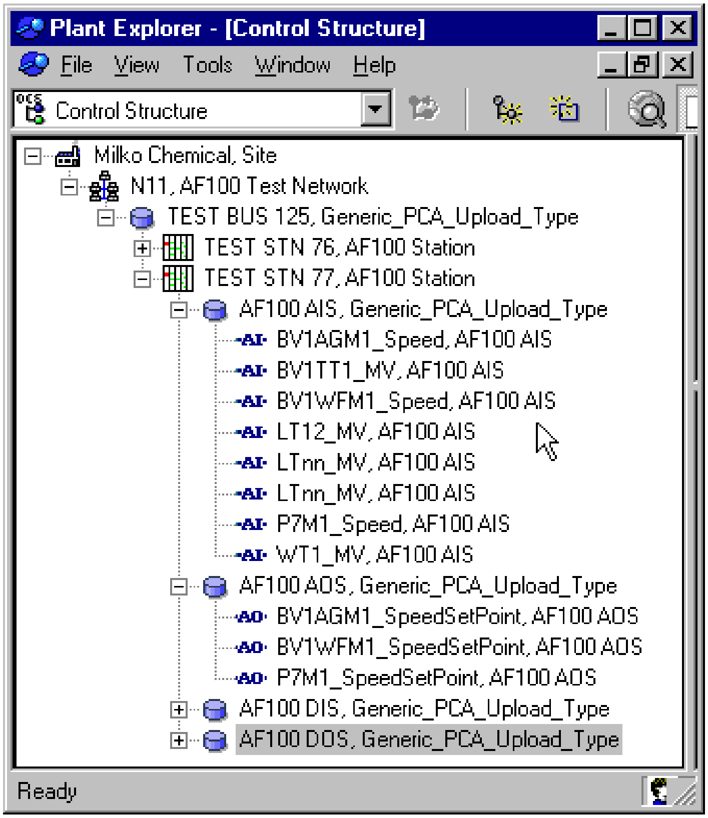


Figure 15-1. Uploader aspect dialog

Typically, the Control structure will now contain the Controllers (Advant System objects that represent the Controller nodes), the object types (Advant System objects representing the group of objects of that type) and the Signal objects.

Figure 15-2 shows an example of the control structure.


Plant Explorer - [Control Structure]
File View Tools Window Help
Control Structure
Ready

- Milko Chemical, Site
  - N11, AF100 Test Network
    - TEST BUS 125, Generic_PCA_Upload_Type
      - TEST STN 76, AF100 Station
      - TEST STN 77, AF100 Station
        - AF100 AIS, Generic_PCA_Upload_Type
          - BV1AGM1_Speed, AF100 AIS
          - BV1TT1_MV, AF100 AIS
          - BV1WFM1_Speed, AF100 AIS
          - LT12_MV, AF100 AIS
          - LTnn_MV, AF100 AIS
          - LTnn_MV, AF100 AIS
          - P7M1_Speed, AF100 AIS
          - WT1_MV, AF100 AIS
        - AF100 AOS, Generic_PCA_Upload_Type
          - BV1AGM1_SpeedSetPoint, AF100 AOS
          - BV1WFM1_SpeedSetPoint, AF100 AOS
          - P7M1_SpeedSetPoint, AF100 AOS
        - AF100 DIS, Generic_PCA_Upload_Type
        - AF100 DOS, Generic_PCA_Upload_Type

Figure 15-2. Example of a control structure after upload

See the relevant sections of Chapter 8, Build the Functional Structure, how to create objects in a structure and how to assign engineering/application data to objects.

15.2 Removing Obsolete Objects

All objects that have disappeared from the Controllers are placed under the **Obsolete Objects** Object under your Network Object. To delete these Objects, you need to perform a **Delete** from the object tree using the Advant Plant Explorer.

15.3 Verifying the Result of an Upload

All objects that are now populated into the Control structure can be subscribed upon, even controlled. If you want to verify that a few of the Signal objects work properly you can use either one of the Graphic element aspects, see Figure 15-3, that exist for that object type or you can use the PCA aspect which allows you to verify the value of all object attributes.

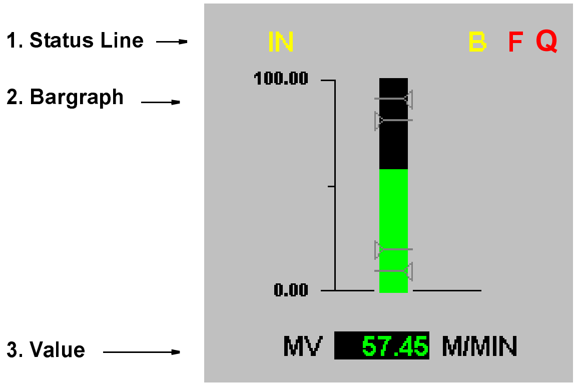
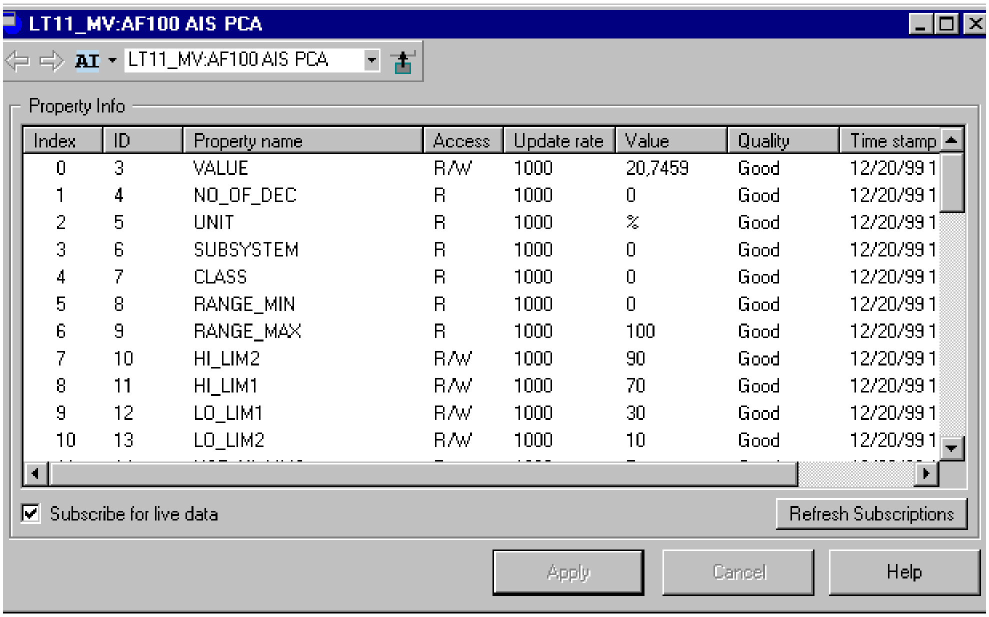


Figure 15-3. Example of a Close-up for an AIS object

A short example on how to use the PCA aspect

PCAs, short for Process Control Aspect, are found as properties on the physical objects. They are found in the **Control structure\ <physical object> \ AF100 “type” PCA**, for example AF100 AIS PCA. Through the PCA, the connection to the OPC server can be verified.

By checking the Subscribe for live data check box, live data is displayed for the PCA. The fields **Value**, **Quality** and **Time Stamp** will start showing live data. When live data can be displayed, the connection to the OPC server is OK.



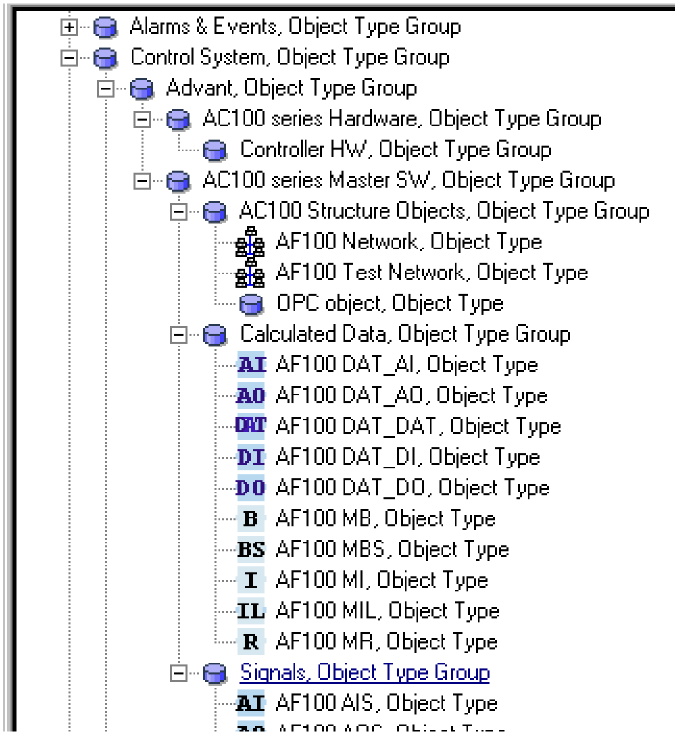
Index	ID	Property name	Access	Update rate	Value	Quality	Time stamp
0	3	VALUE	R/W	1000	20.7459	Good	12/20/99 1
1	4	NO_OF_DEC	R	1000	0	Good	12/20/99 1
2	5	UNIT	R	1000	%	Good	12/20/99 1
3	6	SUBSYSTEM	R	1000	0	Good	12/20/99 1
4	7	CLASS	R	1000	0	Good	12/20/99 1
5	8	RANGE_MIN	R	1000	0	Good	12/20/99 1
6	9	RANGE_MAX	R	1000	100	Good	12/20/99 1
7	10	HI_LIM2	R/W	1000	90	Good	12/20/99 1
8	11	HI_LIM1	R/W	1000	70	Good	12/20/99 1
9	12	LO_LIM1	R/W	1000	30	Good	12/20/99 1
10	13	LO_LIM2	R/W	1000	10	Good	12/20/99 1

☑ Subscribe for live data

Figure 15-4. Live data through the PCA

15.4 Available AC 100 Series Object Types

Figure 15-5 shows all the available AC 100 Series object types in the control structure.



- Alarms & Events, Object Type Group
- Control System, Object Type Group
  - Advant, Object Type Group
    - AC100 series Hardware, Object Type Group
      - Controller HW, Object Type Group
    - AC100 series Master SW, Object Type Group
      - AC100 Structure Objects, Object Type Group
        - AF100 Network, Object Type
        - AF100 Test Network, Object Type
        - OPC object, Object Type
      - Calculated Data, Object Type Group
        - AI AF100 DAT_AI, Object Type
        - AO AF100 DAT_AO, Object Type
        - DAT AF100 DAT_DAT, Object Type
        - DI AF100 DAT_DI, Object Type
        - DO AF100 DAT_DO, Object Type
        - B AF100 MB, Object Type
        - BS AF100 MBS, Object Type
        - I AF100 MI, Object Type
        - IL AF100 MIL, Object Type
        - R AF100 MR, Object Type
      - Signals, Object Type Group
        - AI AF100 AIS, Object Type

Figure 15-5. Object types in Control structure in AC 100 Series release

The following object types exist:

- The Network types: AF100 Network and AF100 Test Network
- Controller Hardware
- Signals - AF100 AIS, AF100 AOS, AF100 DIS, AF100 DOS
- Calculated Data - AF100 DAT_AI, AF100 DAT_AO, AF100 DAT_DAT, AF100 DAT_DI, AF100 DAT_DO, AF100 MB, AF100 MBS, AF100 MI, AF100 ML, AF100 MR.

Chapter 16 Complement the Functional Structure

16.1 Complementing the Functional Structure with Signal Objects

After you built the Functional structure the Control structure is now populated with all Signal objects. This is a good time to complement the Functional structure with the Signal objects that you have not yet entered there.

This is easiest done as a drag-and-drop operation that requires that you start up two Advant Plant Explorer windows.

1. Start up two Advant Plant Explorer windows for example by selecting Tiled from the Advant Plant Explorer's Windows menu and then **File > New**.
2. Select the Functional structure in the first window and the Control structure in the second.
3. Browse the Functional structure to show the object you want as a parent to the first Signal object to show in the Functional structure.
4. Browse the Control structure until you have found the signal object to move.
5. Drag the Object from the Control structure **with <CTRL+SHIFT> pressed** and drop it into the Functional structure at the identified parent. A Functional structure aspect is automatically created but the Signal object will still be visible also in the Control structure.
6. Repeat step 3-5 as many times you need.

For details on how to use the mouse together with or without **<CTRL>**, **<CTRL+SHIFT>** etc., see the Advant Plant Explorer User´s Guide.

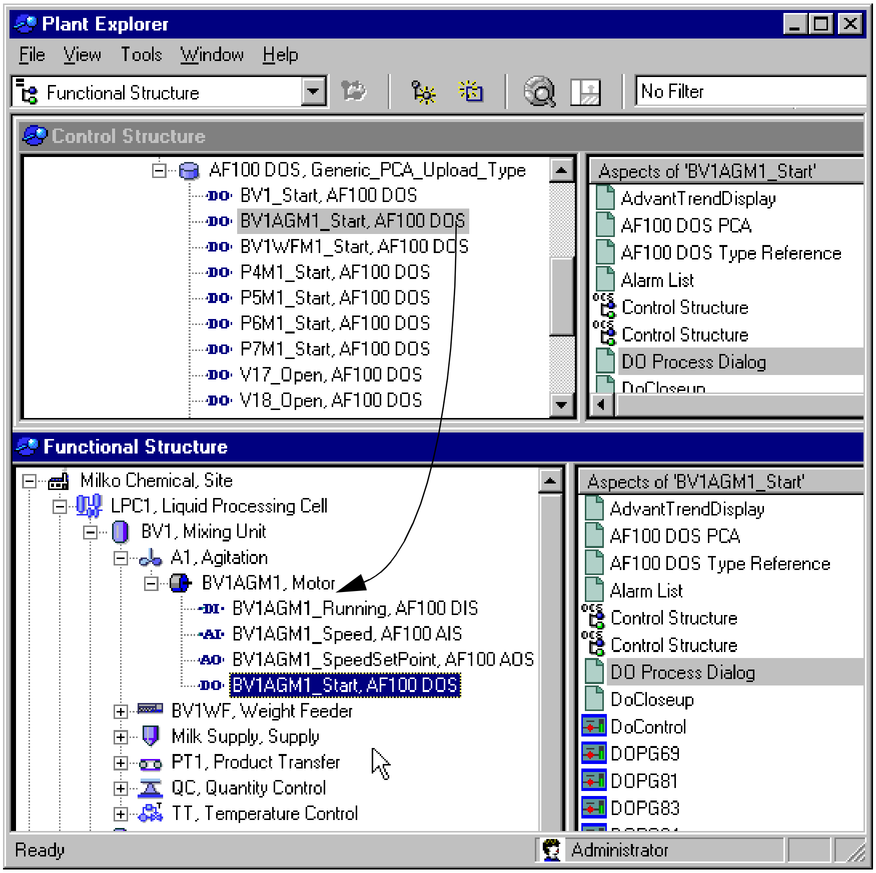


Figure 16-1. Inserting a signal object into the Functional structure

16.2 Recommendations

The Functional structure needs to contain enough Process and Signal Objects to describe the functionality of the plant properly. Normally you use graphic displays to monitor both overview and details and you control the process through dialogs started from the graphic displays. The Functional structure should rather be seen as a Functional overview and a navigation facility. Please note however that in most cases the Functional structure and the graphic displays are two views of the same thing - the process.

There is a good reason to include all signal objects in the Functional structure: If you want to create alarm lists over a certain process section, machine or device, all signal objects belonging to it need to exist in the Functional (or Location) structure. See more about this in Chapter 18, Building Alarm and Event Applications.

Chapter 17 Complement the Location Structure

17.1 Complement the Location Structure with Signal Objects

You should complement the process objects describing the Location structure of your plant with the process and signal objects describing the functionality. Normally you do not need to include all signal objects in the Location structure but some complements are probably required.

This is a drag-and-drop operation that requires that you start up two Advant Plant Explorer windows.

1. Start up two Advant Plant Explorer windows for example by selecting Tiled from the Advant Plant Explorer Windows menu and then **File > New**.
2. Select the Location structure in the first window and the Functional or Control structure in the second.
3. Browse the Location structure to show the object you want as a parent to the first Process or Signal object to show in the Location structure.
4. Browse the Functional or Control structure until you have found the object to move.
5. Drag the Object from the Functional/Control structure **with <CTRL+SHIFT> pressed** and drop it into the Location structure at the identified parent.
   A Location structure aspect is automatically created but the object will still be visible also in the original structure.
6. Repeat step 3-5 as many times you need.

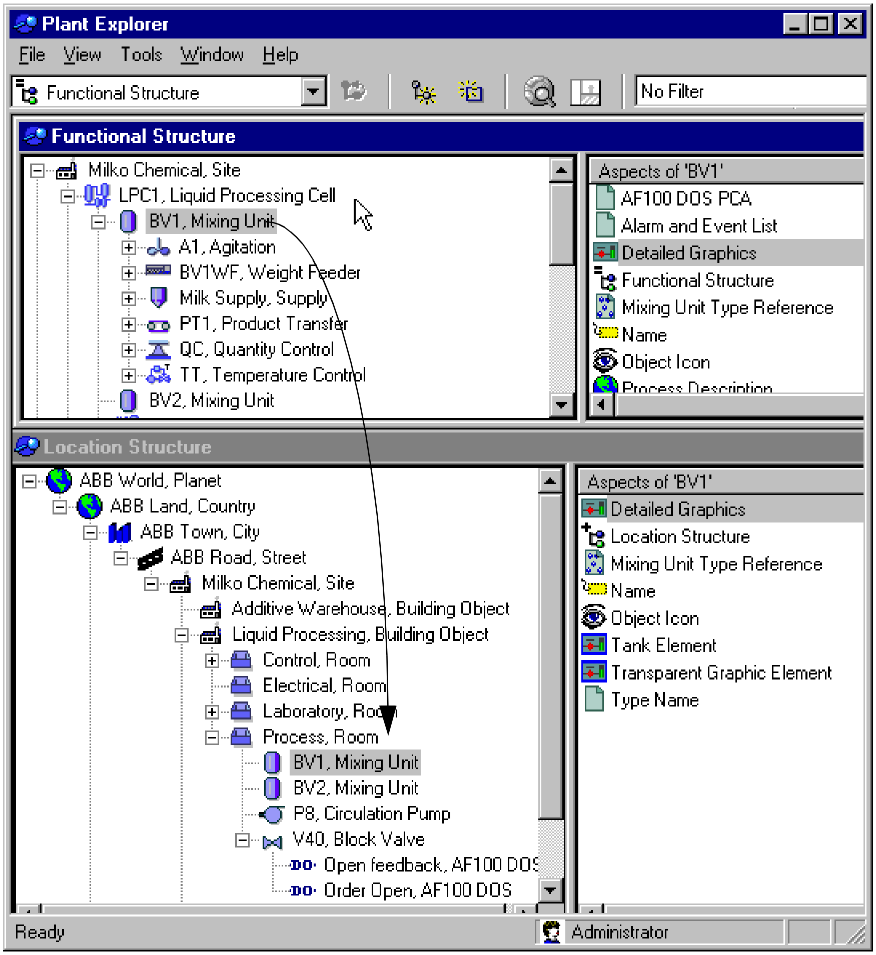


Figure 17-1. Inserting a signal object into the Location structure

17.2 Recommendations

The more complex the Location structure of the site is, the more essential is it that all objects are represented in the Location structure. A complete Location structure gives you a superior navigation. Then the completeness of the Location structure is more a matter of how much work you are willing to spend on completing it.

On the other hand, if the whole site is placed in one room or is very straightforward you might not even need a Location structure.

Figure 17-2 shows a small example of a Location structure.

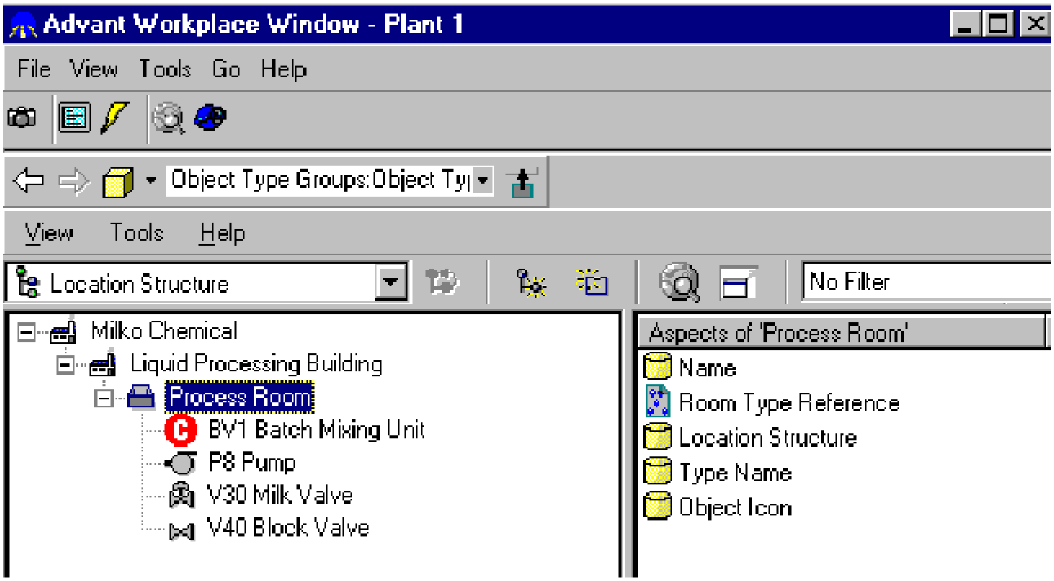


Figure 17-2. A small example of a Location structure

The Location structure needs to contain enough Process and Signal Objects to describe the locations of the important devices properly. The Location structure should be seen as a Location overview and a navigation facility.

You normally do not need to include more than a few signal objects in the Location structure. Important signal objects which may be difficult to find the location for because they have similar names or some other reason should be placed there though.

This recommendation also means that when you want to use the Location structure for an object that does not have a Location structure aspect you just move upwards in the Functional structure until you find a Location structure.

Chapter 18 Building Alarm and Event Applications

18.1 Alarm and Event overview for AF100

Before we describe how to configure the alarm and event client aspects there are some terms and some principles for Advant Fieldbus 100 that need to be described.You should read Section 12.5, More Information on DB Elements before you read this section.

Simplified one can say that for the Advant Controllers of the 100 Series, the signal objects contain one or several Event_Treatment parameters. (Named either ERR_TR, LIM1_TR, LIM2_TR or VALUE_TR.)

1. You assign a value to these Event_Treatment parameters according to Table 18-1.

Table 18-1. Event_Treatment values

Value	Results in
0	no events generated
1	event generated
2	alarm generated(1)
3	event generated for limit transgression
4	alarm generated for limit transgression(1)

(1) An alarm always results in both an alarm and an event in AdvaSoft 2.0 for Windows NT

2. When something occurs for the Signal object, for example a limit transgression, a time stamp and some additional information is sent over in an EVS to the Advant OPC Server for Advant Fieldbus 100.
3. The OPC Server translates the information, to among other things, a severity number and to a corresponding event text. For details on this and on how you

could change this translation, see Appendix C in the *Advant OPC Server for Advant Fieldbus 100 User´s Guide*, 3BSE013896.

4. The resulting event and alarm information is distributed to AdvaSoft 2.0 for Windows NT where you can see the results in the event and alarm aspects.

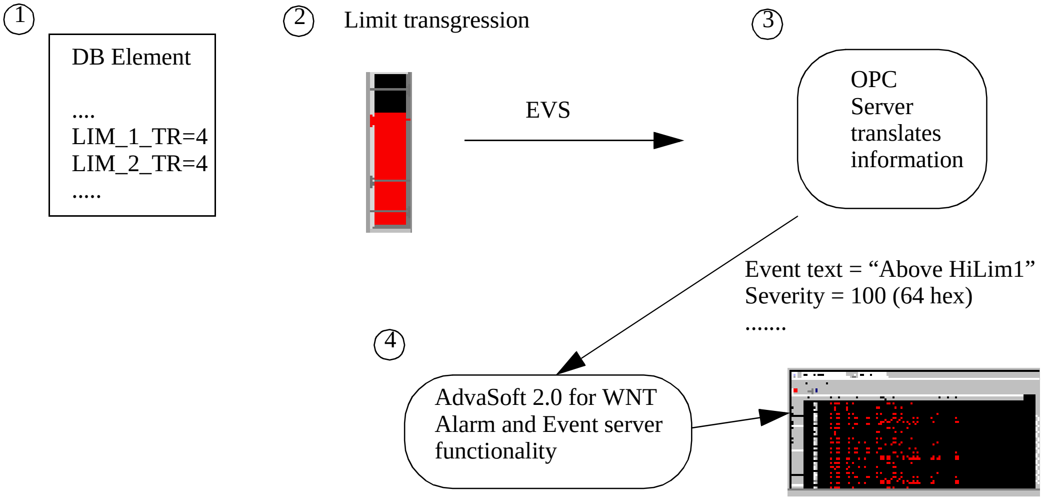


Figure 18-1. Principles for Alarm and Event distribution

The Severity is an OPC terminology. You can have severities between 1-1000. The Alarm and Event list aspect uses the term priority for severity. It divides the severity levels into a number of priority intervals, see Figure 18-3.

The current versions of the Engineering tools and OPC Server for Advant Fieldbus 100 do not support severity levels per object. Neither does it distinguish between AIS and AOS object types. Thus, you can not use several priority intervals in a consistent way yet. The current translation of the Event information places the severity for all events in a range between 98 and 102. With 10 priority intervals some alarms will get priority 2 and some priority 1. With fewer intervals all alarms will get priority 1. See Appendix C in the *Advant OPC Server for Advant Fieldbus 100 User´s Guide* for details.

18.2 Alarm and Event defaults

For your convenience the Alarm and Event functionality provides you with some default functionality.

The Alarm and Event List aspect presents alarm and/or events according to a configuration that you select. There is always a Default alarm list configuration and a Default event list configuration that you can select. You can also create your own default configurations giving your alarm and event presentations your own touch. You can customize filtering, presentation layout and size, colors, time and date format etc. As a start we recommend that you use the existing default configurations and learn more about all the various possibilities that exist before you start to do any alterations. So, you can continue by reading Section 18.3, Creating full Alarm and Event Lists and return back to this section when you understand more.

Here we assume that you know a lot about the alarm and event functionality and are ready to do your own configurations for the Alarm and Event list aspects.

So, let us describe how to accomplish this. You want to create a Default configuration for the alarm list with the configuration changed (compared to the default) as follows:

- The Comment column added to the right wing of the presentation
- A specified list of additional columns to present.
- The number of presented alarms 150.
- All alarms are red and all acknowledged alarms (Ack) are yellow
- A momentary ringing shall sound when an alarm occurs.

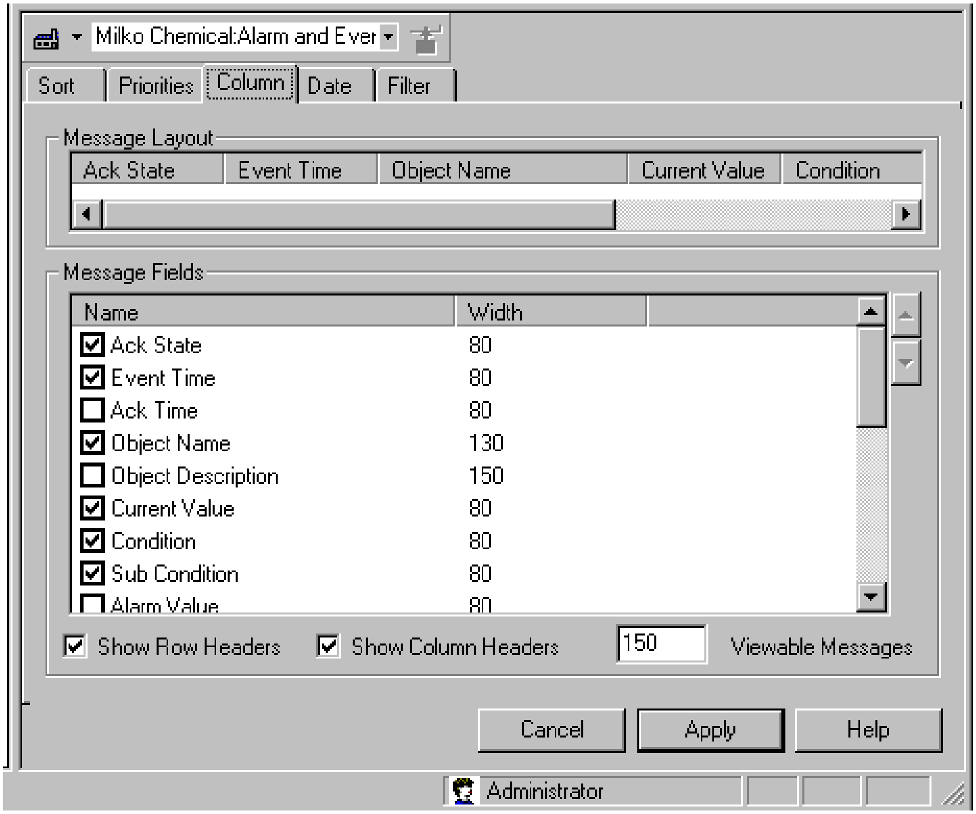
The details of each of the required configuration actions is described in the *Advant Alarms User´s Guide*. However, here you will get a brief outline how to do this.

First, lets create a Default configuration aspect for our alarm lists.

1. Select the Library Structure.
2. Add a new Alarm & Event List Configuration object to the existing defaults by selecting Alarm and Event List Configurations.
    a. Select Alarm and Event List Configurations object.
    b. Select New Object.

   c. Select Alarm & Event List Configuration.
   d. Name it Default Alarm List Milko.

3. Select the new object and double click the Alarm and Event List Configuration aspect within the object. This opens the Config View menus.
4. Click on the **Filter** tab and change the Message Type to **Process Alarms**.
5. Click on the **Column** tab and make the following changes.



Message Layout

Ack State	Event Time	Object Name	Current Value	Condition

Message Fields

Name	Width
☑ Ack State	80
☑ Event Time	80
☐ Ack Time	80
☑ Object Name	130
☐ Object Description	150
☑ Current Value	80
☑ Condition	80
☑ Sub Condition	80
☐ Alarm Value	80

☑ Show Row Headers   ☑ Show Column Headers   150 Viewable Messages

Figure 18-2. Changing the Alarm List Column configuration

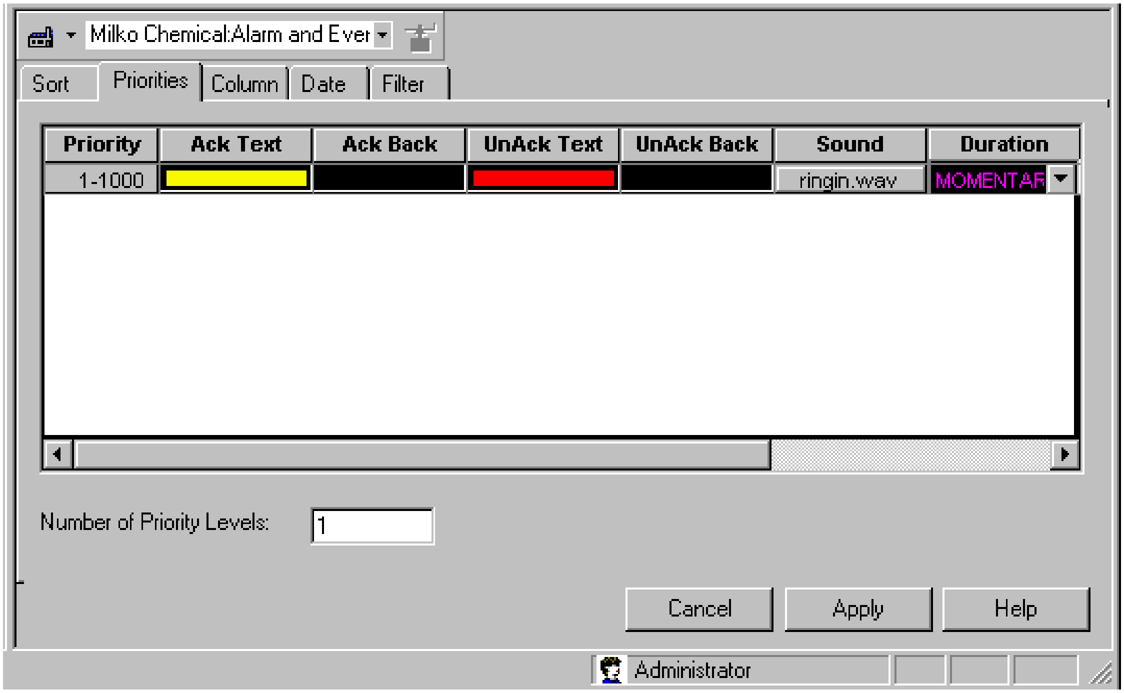
   a. Make sure that the following columns are checked by checking and unchecking check boxes: Ack State, Event Time, Object Name, Current Value, Condition, Sub Condition and Comment.

   b. Adjust the width of the Object Name column to 130, change the number of viewable messages to 150 and study the result. Click on **Apply**

6. Now you want to rearrange so that all alarms are red and all acknowledged alarms (Ack) are yellow and you want a momentary ringing to sound when an alarm occurs.

   a. Click on the **Priorities** tab.

   b. Select 1 priority level, see Section 18.1, Alarm and Event overview for AF100, for details why.



Priority	Ack Text	Ack Back	UnAck Text	UnAck Back	Sound	Duration
1-1000	🟨	⬛	🟥	⬛	ringin.wav	MOMENTAR

Number of Priority Levels: 1

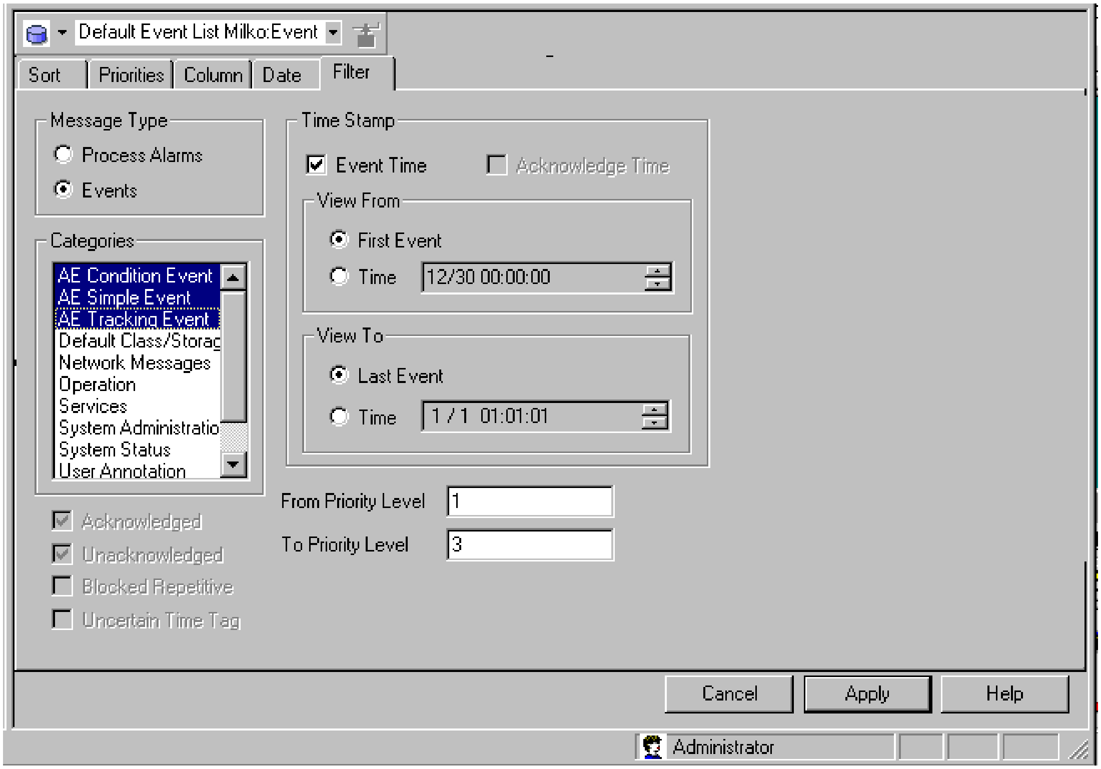
Figure 18-3. Changing the Alarm List Priority configuration

   c. Click on the text color field for acknowledged alarms (Ack Text) and select yellow. Repeat with the text color field for unacknowledged alarms (UnAck Text) and select red.

   d. Click on the Sound box and browse to a suitable sound. Select Momentary duration. Click on **Apply**.

7. Close the configuration menus (Alt+F4) or click the top right corner.

Secondly, let us create a Default Event List Configuration for Milko.



Default Event List Milko:Event

Message Type
○ Process Alarms
● Events

Categories
AE Condition Event
AE Simple Event
AE Tracking Event
Default Class/Storag
Network Messages
Operation
Services
System Administratio
System Status
User Annotation

☑ Acknowledged
☑ Unacknowledged
☐ Blocked Repetitive
☐ Uncertain Time Tag

Time Stamp
☑ Event Time ☐ Acknowledge Time

View From
● First Event
○ Time 12/30 00:00:00

View To
● Last Event
○ Time 1 / 1 01:01:01

From Priority Level 1
To Priority Level 3

Figure 18-4. Event List Filter configuration

Do the necessary steps to configure the event list according to the needs for the Milko Milkshake site. This is very similar to setting up the alarm list as described above.

a. Priorities: No sound file.

b. Columns: Remove the Ack State and Current Value columns

c. Filter: Configure for events instead. Select Event categories according to Figure 18-4.

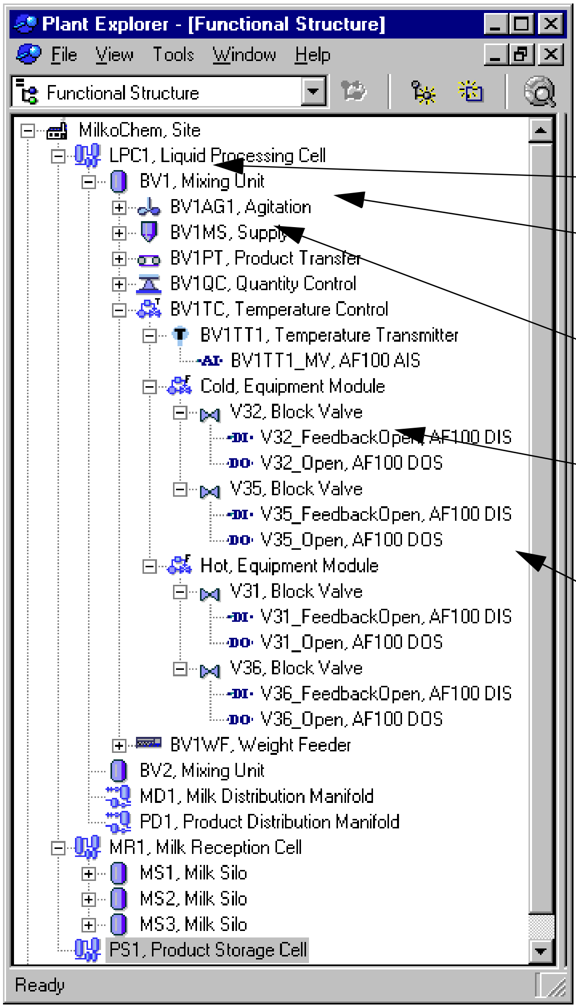
18.3 Creating full Alarm and Event Lists

The same applies for Alarm and Event aspects as for graphic displays. You probably want the Alarm list and corresponding Event List organized according to your Functional structure.

At the top object for the site you place Alarm and Event Lists that cover the whole site. Then, for each process cell (if your type of process is divided in cells, you may call it sections or other things) you will probably have an alarm list, perhaps also an event list. As you can see in Figure 18-5, you can continue to break the alarm and event information down in smaller and smaller entities until you are down on the Signal object level. This is very similar to how you break down the graphical presentation.

For each logical entity you can then see not only a graphical presentation, you can also have an alarm list with the alarms from all the object from that entity (ABB object).

Now, let us see how we build this up.


Plant Explorer - [Functional Structure]
File View Tools Window Help
Functional Structure
Ready

- MilkoChem, Site
  - LPC1, Liquid Processing Cell
    - BV1, Mixing Unit
      - BV1AG1, Agitation
      - BV1MS, Supply
      - BV1PT, Product Transfer
      - BV1QC, Quantity Control
      - BV1TC, Temperature Control
        - BV1TT1, Temperature Transmitter
          - BV1TT1_MV, AF100 AIS
        - Cold, Equipment Module
          - V32, Block Valve
            - V32_FeedbackOpen, AF100 DIS
            - V32_Open, AF100 DOS
          - V35, Block Valve
            - V35_FeedbackOpen, AF100 DIS
            - V35_Open, AF100 DOS
        - Hot, Equipment Module
          - V31, Block Valve
            - V31_FeedbackOpen, AF100 DIS
            - V31_Open, AF100 DOS
          - V36, Block Valve
            - V36_FeedbackOpen, AF100 DIS
            - V36_Open, AF100 DOS
      - BV1WF, Weight Feeder
    - BV2, Mixing Unit
    - MD1, Milk Distribution Manifold
    - PD1, Product Distribution Manifold
  - MR1, Milk Reception Cell
    - MS1, Milk Silo
    - MS2, Milk Silo
    - MS3, Milk Silo
  - PS1, Product Storage Cell

Full alarm and event lists

A Process Cell with an alarm list

A unit with it´s own alarm list

An equipment module alarm list

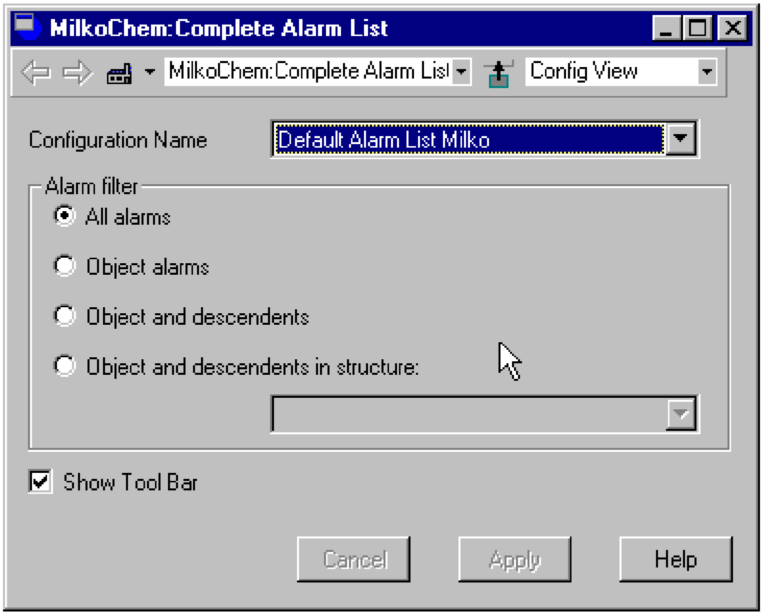
Signal object alarm list

Figure 18-5. Organizing Alarm and Event List

You probably want to have a complete alarm list showing all alarms placed on one of the top objects representing the whole plant. You also want a complete Event list placed accordingly. Let’s select the **Milko Chemical** object which is the object representing the whole site.

Then perform the following steps to create the alarm and the event lists respectively.

1. Start Advant Plant Explorer if not already running.
2. Select the Functional structure.
3. Select the **Milko Chemical** object.
4. Create an Alarm and Event List aspect with the name Complete Alarm List.
5. Create an Alarm and Event List aspect with the name Complete Event List.



MilkoChem:Complete Alarm List

MilkoChem:Complete Alarm List | Config View

Configuration Name: Default Alarm List Milko

Alarm filter
- ● All alarms
- ○ Object alarms
- ○ Object and descendents
- ○ Object and descendents in structure:

☑ Show Tool Bar

Figure 18-6. Alarm List Config View

6. Now, right click on the Complete Alarm List and select the Config View. Select the Default Alarm List Milko and check the All alarms checkbox. See Figure 18-6.
7. Do the corresponding configuration for the Complete Event List. Just select the Default Event List Milko and check the All alarms checkbox.

In the next chapter, Chapter 19, Customizing User Environment, Advant Workplaces, you will se how to use the alarm band aspect so that you can present the alarm or event list with just one mouse click.

Before doing so we will create some customized alarm lists that each one describe a a process cell, making it easier to quickly identify what is wrong in each respective cell.

18.4 Creating Alarm Lists for Important Plant Sections

To create alarm aspects for an important plant section:

1. Start Advant Plant Explorer if not already running.
2. Select the Functional structure.
3. Select an object representing a process cell or another important section, start with the BV1.
4. Create an Alarm List aspect as described in the previous tutorial. Name it “Process Cell Alarm List”.
5. Place the cursor on the aspect and right click. Select the Config View.
6. Check the “Objects and descendents in structure” and select the Functional structure.
7. Select Default Alarm List Milko in the Configuration Name so we get the same configuration with regards to colors and columns as for the overview alarm list. Click on **Apply**.
8. Repeat step 3-7 for each one of the other important sections.

In Chapter 19, Customizing User Environment, Advant Workplaces, you will se how we build an alarm band to these alarm aspects so that you can present the alarm list with just one mouse click.

18.5 Configuring the Alarm Band

As the Alarm Band functionality has a lot of relations with how you customize your Advant Workplace it is described in Chapter 19, Customizing User Environment, Advant Workplaces.

18.6 Configuring Alarm Logging

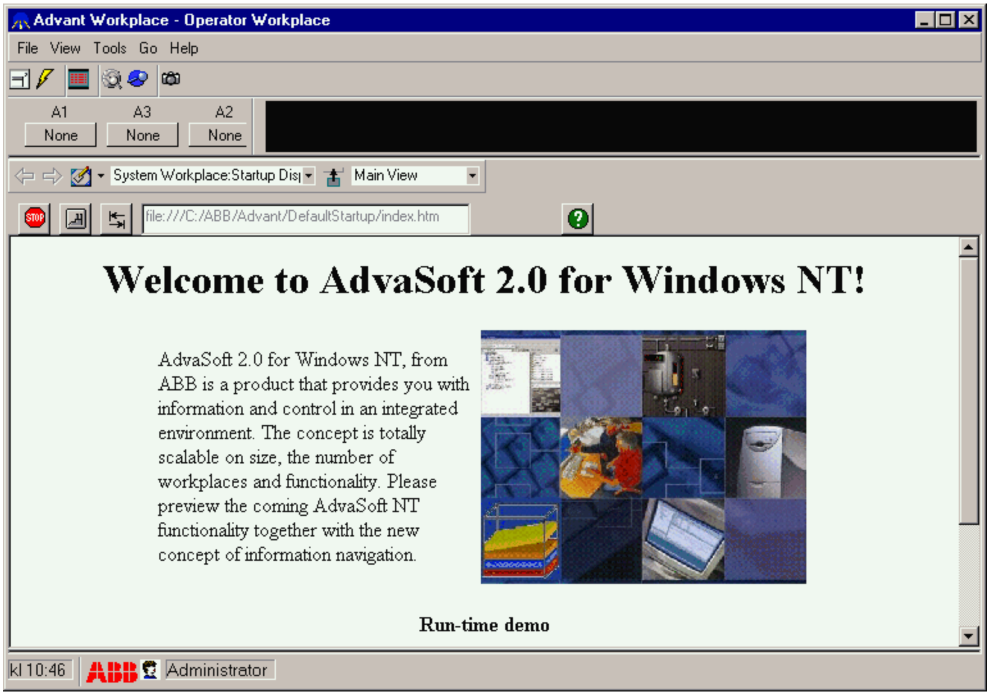
You have decided to have one printer to function as alarm printer. All alarms will be logged on it. You assign the alarm logger aspect to the **Milko Chemical** Object and use the Printer AlarmLogprinter (which was assigned according to Windows NT instructions).

To create an alarm logger aspect:

1. Start Advant Plant Explorer if not already running.
2. Select the Functional structure.
3. Select the **Milko Chemical** object.
4. Click on the new aspect icon, browse to the Alarm Logger aspect and then click on **Create** to create the alarm logger aspect.
5. Click on the aspect to edit it. Select the log printer to be Alarmlogprinter.
6. If you want to change the printout format you could create your own configuration aspect as for the alarm list but in this case we just select the default alarm logger configuration.
7. Click on **Start**. Wait for about 15 seconds and then click on Get Logger Status to see if it started properly.
8. Now you can also see alarms being printed on the printer as they occur.

Chapter 19 Customizing User Environment, Advant Workplaces

19.1 Customizing the User Environment


Advant Workplace - Operator Workplace
File View Tools Go Help
A1 A3 A2
None None None
System Workplace:Startup Dis|
Main View
STOP
file:///C:/ABB/Advant/DefaultStartup/index.htm
kl 10:46
ABB
Administrator

Welcome to AdvaSoft 2.0 for Windows NT!

AdvaSoft 2.0 for Windows NT, from ABB is a product that provides you with information and control in an integrated environment. The concept is totally scalable on size, the number of workplaces and functionality. Please preview the coming AdvaSoft NT functionality together with the new concept of information navigation.

Run-time demo

Figure 19-1. The default Workplace set-up

You have a very powerful set of tools with which you can customize the Advant Workplace - the user environment. You can

- configure the Application Bar
- configure the Status Bar

- build fixed displays
- build Panels
- create display short-cuts to ease navigation
- create Alarm bands - short-cuts to Alarm and Event Lists
- create your own menu displays
- set up a start-up display
- set up the start-up mode
- setting up the final users

The details on how to do this are described in the *Advant Workplace User´s Guide*. You can find a figure describing the Advant Workplace Window and it´s terminology in Section 1.6.3, Layout of the Advant Workplace Window.

In this document we will only describe briefly how to configure the **Application Bar** or the **Status Bar**. You already have a default configuration of those that will suite most needs and includes the inherited knowledge of Operator Workplaces that ABB´s own engineers have. The application bar already has a **fixed display** containing a two line alarm list and another containing an Alarm band.

Aspect views can be displayed as base displays, overlapping displays, part of a **panel** or in a fixed display area. Building panels is a possible way to optimize and customize the way information is presented. See the *Advant Workplace User´s Guide* for details.

This chapter will describe how to set up **display shortcuts -** for the Workplace or just for aspects of dedicated objects. It will also describe how to:

- create Alarm bands
- create your own menu displays
- setting up the final users

You can read the *Advant Workplace User´s Guide for details on how to* set up a **start-up display**. To learn how to set up the **start-up mode** use the *Advant Workplace Products Administrators Guide.*

19.2 Set-up of Direct Navigation to Overview Displays

Now the graphic displays and alarm lists for the whole site as well as for the process cells exist. As a lot of the work from an operator perspective is done from an overall or from a process cell perspective we want to have direct access to those important displays.

The Advant Workplace has two areas within which you can place direct access icons, the Application Bar and the Display Bar. The Application Bar is always included in the Advant Workplace while the Display shortcuts in the display bar have a local meaning. See also Figure 1-2.

19.2.1 Display Shortcuts in the Advant Workplace

In the Advant Workplace (in the Application bar) we want to place a display shortcut to the overview graphic display and to some of the graphic displays on the process cells.

Follow these steps to include the overview graphic display:

1. Open the Advant Plant Explorer and select the Workplace structure
2. Go to the Workplace Object, to which you want to include a Display Shortcut.
3. **Create a New Aspect** of the Application Bar Shortcuts” aspect category.
4. Double click on the aspect, and a dialog box opens.
5. Click the **Add** button, and a dialog box pops up.
6. Choose the relevant Functional Structure, the Milko Chemical Object and the Overview Graphics aspect in the dialog box.

   Choose “Presentation mode” Base, because you want the Display Shortcut to give you the defined new display in the Base Display Area. Click on the Apply button.

   The Display Shortcut link is added to the Display Shortcut list.
7. Repeat step 5-6 for more aspects.
8. Click on the **OK** button, and the Display Shortcut is created in the Application Bar.

ℹ If the chosen Advant Workplace Window is open, you have to close it and open it again for the chosen changes to take effect.

The Display Shortcut buttons will look like the default icons of the referenced objects. If they point to aspects of the same object they will look the same.

If you want the steps in more detail and with more visualization please see *Advant Workplace User's Guide*.

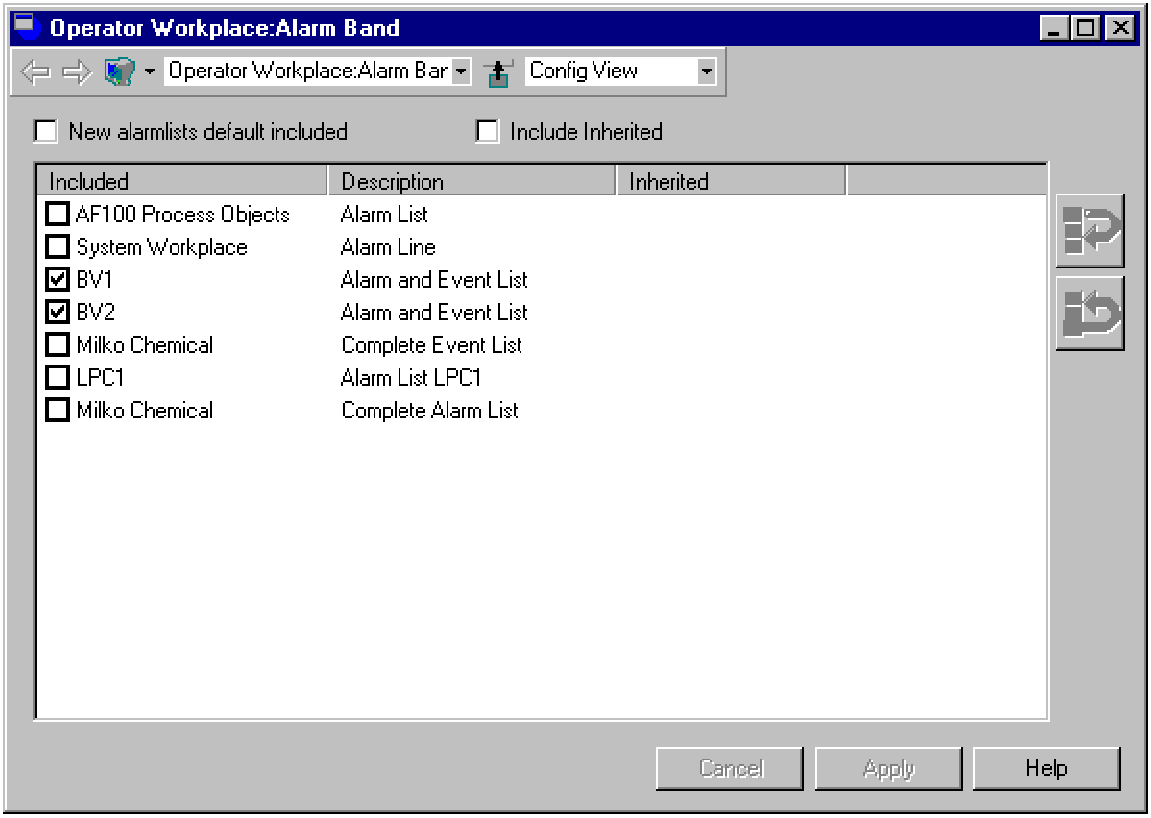
19.2.2 Configuring Alarm Bands - short-cuts to Alarm and Event lists

An alternative to using application bar shortcuts to an alarm or event list aspect is to use the Alarm band aspects. The Alarm Band aspect makes it possible to select a number of alarm lists to be represented in a list of Alarm icons, each icon representing the number of alarms in that specific alarm or event list. A default set up of an Alarm Band exists in the default Operator Workplace. To place it in a Workplace where it does not yet exist you have to add it as a fixed display. See the *Advant Workplace User´s Guide*

In the default Workplace an empty Functional structure exists with three area objects, A1, A2, A3. An alarm band including their respective alarm lists are placed as a fixed display in the default workplace, see Figure 19-1.

This is how you do to re-configure the alarm band aspect.

1. Start Advant Plant Explorer.
2. Select the Workplace object from the Workplace structure.
3. Right click on the Alarm Band aspect and select the configuration view. You will then see the current alarm and event list aspects. This does not include the inherited alarm list aspects placed on Signal objects (unless you check the Include inherited box, don't). See Figure 19-2.
4. Click on the Alarm and Event List aspects that you want to include.



☐ New alarmlists default included ☐ Include Inherited

Included	Description	Inherited
☐ AF100 Process Objects	Alarm List	
☐ System Workplace	Alarm Line	
☑ BV1	Alarm and Event List	
☑ BV2	Alarm and Event List	
☐ Milko Chemical	Complete Event List	
☐ LPC1	Alarm List LPC1	
☐ Milko Chemical	Complete Alarm List	

Figure 19-2. Alarm band config view

In this case we select the BV1 and BV2 alarm list and the Milko Chemical Alarm list. You can also rearrange the order of them using the “arrow” push buttons to the right.

See Figure 19-3 for an idea of how the symbols look like in your final display. As soon as any alarm appears in any of those alarm lists the symbol will show it with a number and a highlighted color.

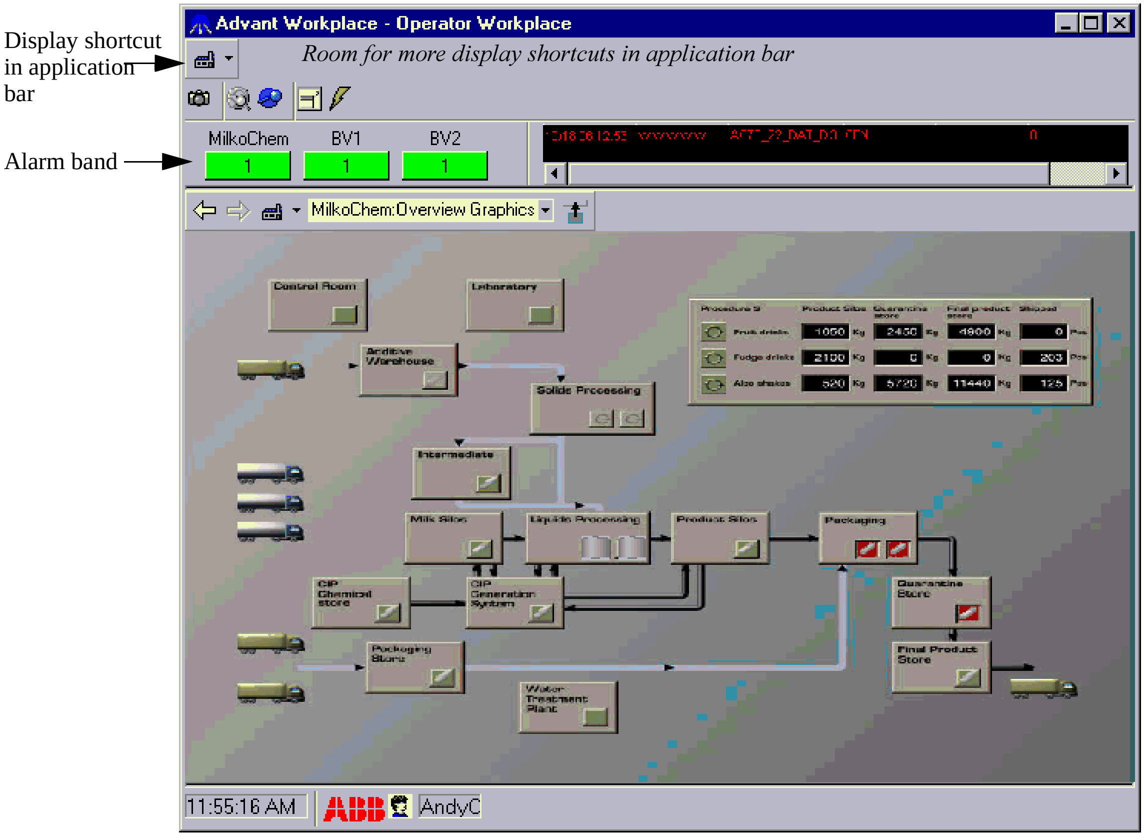


Figure 19-3. Presenting the Alarm Band

19.2.3 Display Shortcuts in the display bar

A Shortcut in the Display Bar is a shortcut button, that you create to get fast access to **any** viewable aspect associated with a particular **ABB Object**. Compare with adding it to the Application Bar where it applies to the Advant Workplace, no matter which display you have selected. When you have added a Display Shortcut to an ABB Object, it will be displayed as a button in the Displays Display Bar.

An important feature of the Display shortcuts placed on an Process object is the inheritance feature. The Display shortcut can be inherited by the underlying objects (default). Furthermore, if you have several objects above, each with display shortcuts, they can all be inherited into the Display bar giving the object in the end of the branch a lot of direct navigation buttons. See the *Advant Workplace User's Guide* for details.

Follow the steps below to create the Display shortcuts for the Process Cell objects:

1. Open Advant Plant Explorer, select a structure (normally Functional).
2. Select the Object, to which you want to add a Display Shortcut.
3. Create a Shortcut aspect
4. Double click on the aspect, and a dialog box opens.
5. Click the Add button, and a new dialog box pops up.
6. Choose the Functional structure, the correct Process Cell Object, and select Aspect in the dialog box. (Remember we want to have direct access to the graphic displays and the Alarm list for the Process Cell. Later we will also add a Trend overview for the object.)

   Choose "Presentation mode" Overlap, as we want a click on the Display Shortcut to give you the new display as an overlapping display on the screen.
7. Click on the **OK** button, and the chosen Shortcut link is added to the Shortcut list.
8. A Shortcut in the Display Bar is created. In Figure 19-4 the overlaps presented when the display shortcuts are pushed are presented in the same screen capture.

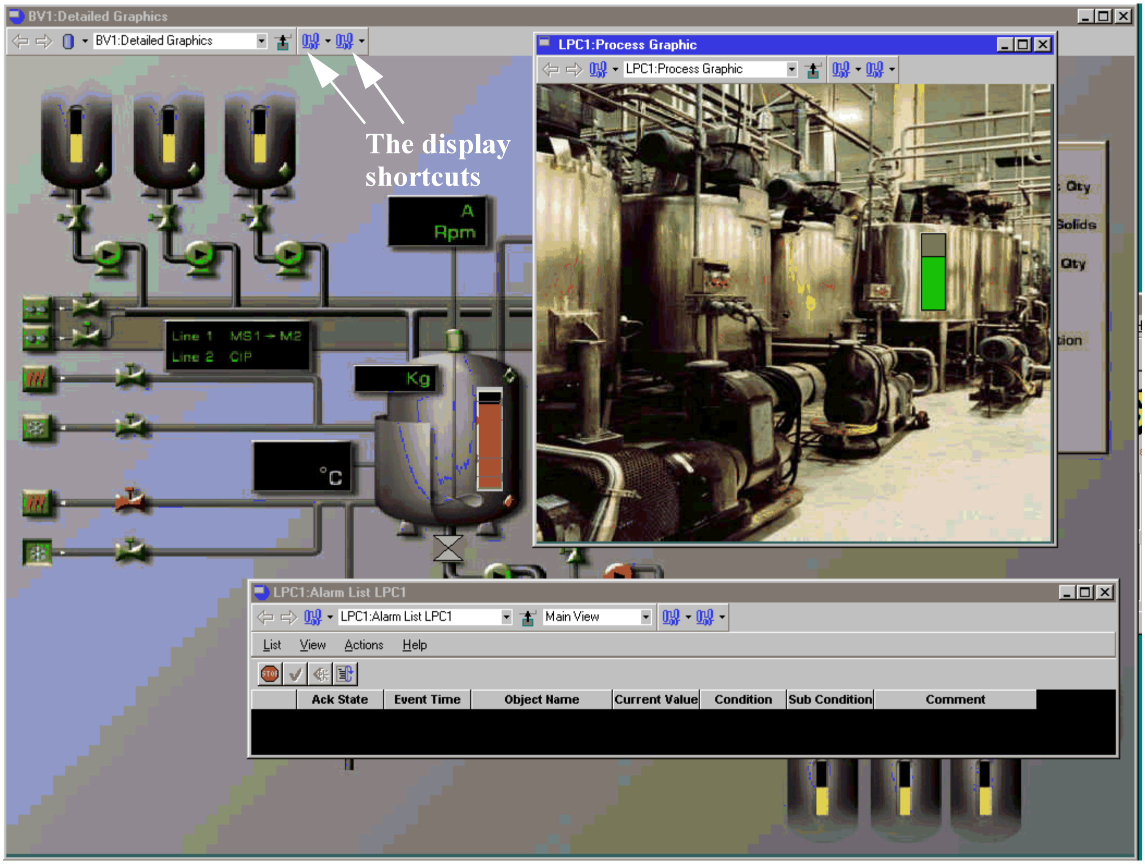


Figure 19-4. A graphical display presented with inherited display shortcuts

You have to close the Advant Workplace Window and open it again for the changes to take effect.

The Shortcut buttons will look like the default icons of the referenced objects.

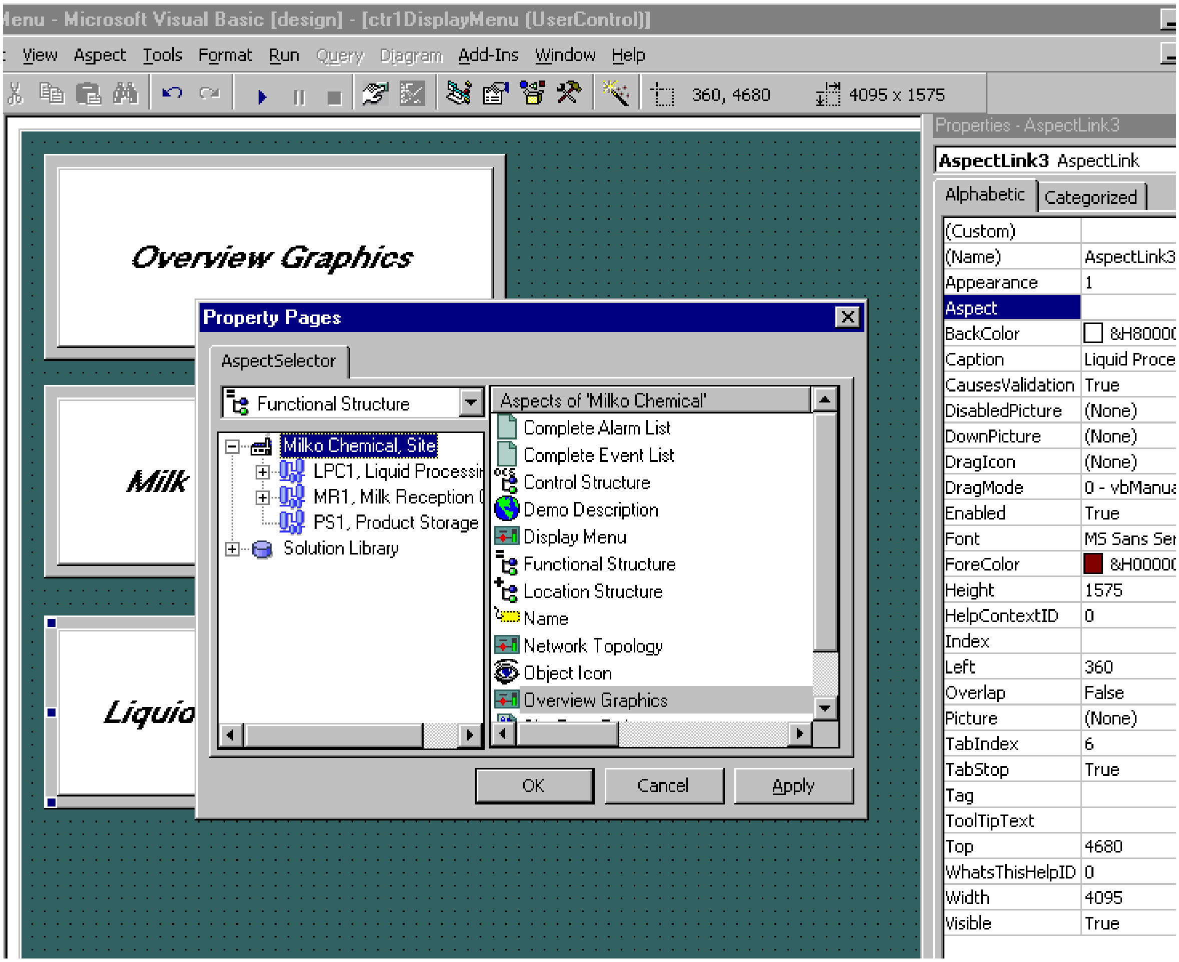
Click the Shortcut Button on the Display Bar to use the Shortcut.

19.3 Building up Your Own Menu Display

The traditional way of navigation has often included navigation using a “display menu”. If you have operators who prefer to navigate using such a menu it can easily be built using the Aspect link graphic element.

Just follow the following steps:

1. Create a graphic display according to the tutorial in *Advant Graphics User’s Guide* (you can have a look at Section 10.3, Building the Milko Overview Graphic Display in this book as well).
2. Build relevant background symbols for the menu items.(In the example we just chose a VB Command button for each one of them.
3. Click the icon for the Aspect Link element in the toolbox. The cursor changes to a crosshair.
4. Place the cursor at any point in Visual Basic’s workspace. (The first time over the symbol for the whole Milko Chemical plant, then for the respective process cell).
5. Click and hold the left mouse button.
6. Drag the cursor diagonally.
7. Release the mouse button. The aspect link now appears inside a bounding rectangle in the workspace
8. Point at the aspect link element with the cursor and then select Properties from the menu that a right click will give you.
9. Fill in the properties. The first time you want the object to be Milko Chemical and the aspect the overview graphic display. Select relevant font and color properties as well.



(Custom)	
(Name)	AspectLink3
Appearance	1
Aspect	
BackColor	8H8000C
Caption	Liquid Proce
CausesValidation	True
DisabledPicture	(None)
DownPicture	(None)
DragIcon	(None)
DragMode	0 - vbManua
Enabled	True
Font	MS Sans Ser
ForeColor	8H00000
Height	1575
HelpContextID	0
Index	
Left	360
Overlap	False
Picture	(None)
TabIndex	6
TabStop	True
Tag	
ToolTipText	
Top	4680
WhatsThisHelpID	0
Width	4095
Visible	True

Figure 19-5. Building the Menu in Graphic Builder, selecting aspect to show

10. Repeats step 3-9 for each one of the menu items.The result can be seen below in Figure 19-6.

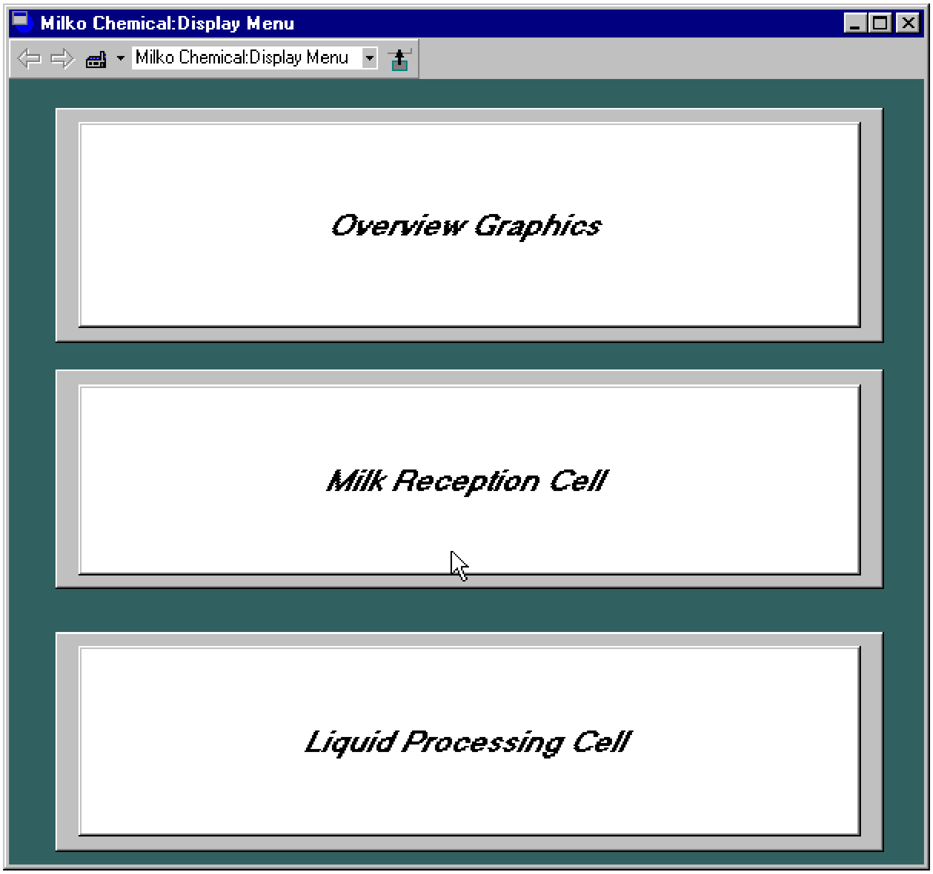


Figure 19-6. A resulting display menu

You should be aware of that any viewable aspect can be chosen as aspect to such a menu. You should also understand that this menu does not need to be built, it is just another alternative way to build up navigation.

19.4 Setting up the Final Users

19.4.1 General

For our Milko Chemical plant we have five operators, three engineers and two maintenance persons. We also want two system administration roles. Some of these users already exist. Let’s look at a table on what we want to do before doing it:

Table 19-1. Users in the Plant

User	Roles	Exists already	Comment
John Ericsson	System Administrator Engineer	Yes	Sys Admin
Andy Cole	System Administrator Engineer	No Yes	Engineer (helping hand sys admin)
David Beckham	Engineer	Yes	Engineer
Jesper Blomqvist	Engineer	No	Engineer
Christian Vieri	Engineer Operator	No	Maintenance
Michael Owens	Engineer Operator	No	Maintenance
GianLuca Vialli	Operator	No	Operator
Pablo Maldini	Operator	No	Operator
Roberto Mancini	Operator	No	Operator
Alan Shearer	Operator	No	Operator
Kevin Keegan	Operator	No	Operator

19.4.2 Creating NT Users

Most of the users already exist as NT users but Pablo Maldini and Roberto Mancini need to be added. This is done as the Windows NT system administrator role and using standard NT tools.

Remember that all advant users need to be assigned to the Advant User group and all advant user's with administrators privileges need to be assigned to the Advant Admin group.

19.4.3 Creating the Advant Users

An Advant user, for example an operator or an application engineer, performs different operations. Advant users can be placed into one or more Advant user groups. By placing Advant users into different user groups they automatically get different privileges.

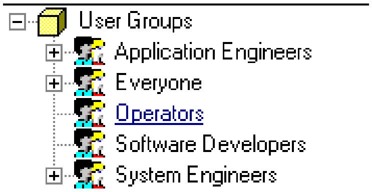


Figure 19-7. User Structure in Plant Explorer

A Windows NT user, which is used when logging into a node, can automatically be promoted to an Advant user. The user and user group concept is used to handle personal customizing, authority control and operation logging of what the users have been doing in the system.

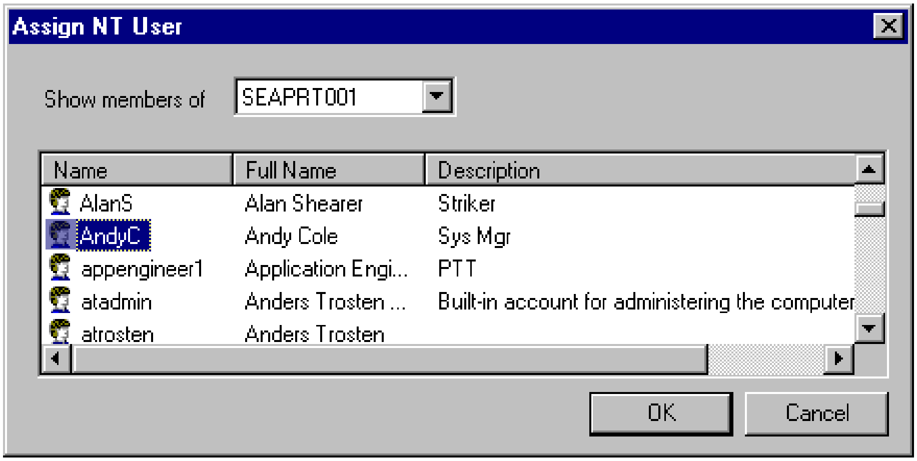
(You can also use the Configuration Wizard to add new users. See the *Advant Workplace Products User´s Guide).* To remove users however, you can only use the Advant Plant Explorer.

All Advant User can be found in the User structure in Advant Plant Explorer.

To add a user to the system:

1. Open Advant Plant Explorer and select the User structure. Select the user group to which you want to add the user.

2. Add a new object to any user group by first selecting the group and then clicking the **New Object** button. A dialog will be displayed, prompting you to enter a name and a description for the user object.
3. Enter a name and a short description.
4. Click **OK** and then **Create** to create the user object.
5. To configure the newly created Advant user object, select the object and click the **User Definition aspect** to open the Definition dialog.
6. On the Identification Tab, the name and the description of the Advant user object can be modified. Click **Apply** to apply the changes.
7. Select the User configuration tab to configure the Advant User Account. Every Advant user has to have a Windows NT account mapped to its Advant user account. To assign an NT account to the Advant User Account, click the **Assign NT Account...** button.



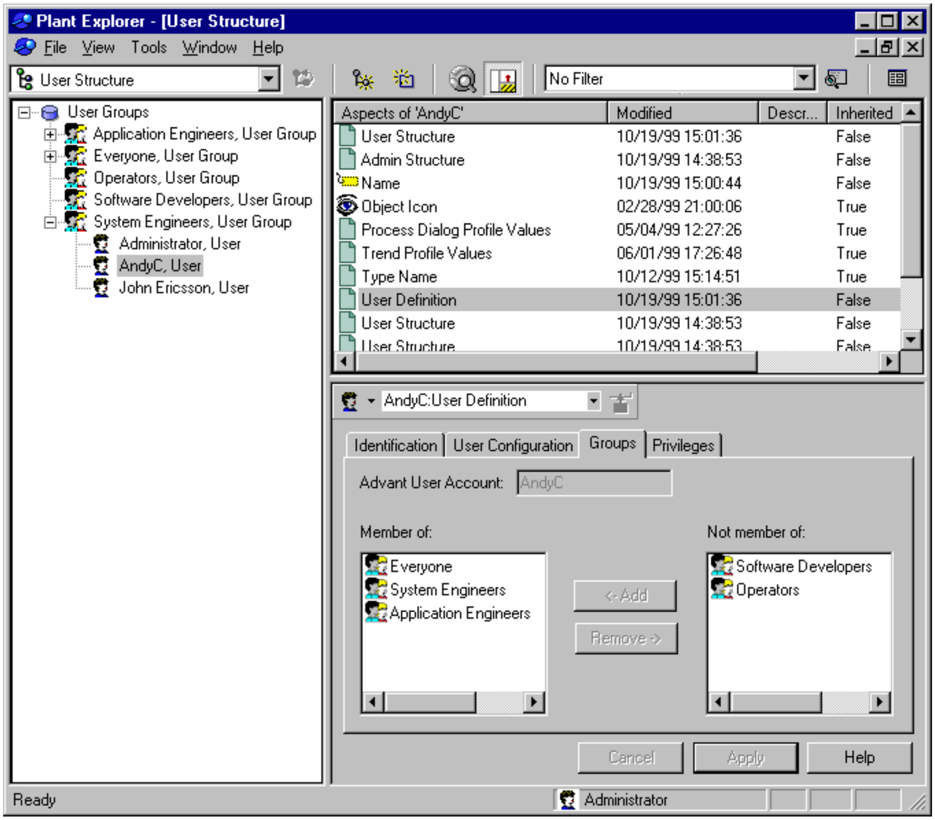
Show members of: SEAPRT001

Name	Full Name	Description
AlanS	Alan Shearer	Striker
AndyC	Andy Cole	Sys Mgr
appengineer1	Application Engi...	PTT
atadmin	Anders Trosten ...	Built-in account for administering the computer
atrosten	Anders Trosten	

Figure 19-8. Assign NT Account

8. The **Assign NT account..**. dialog displays a list of available domains and users as you can see in Figure 19-8. Select the correct NT domain and then the User Account and press **OK** to assign the NT account to the Advant user.

9. The Groups tab displays the user groups to which the current Advant user belongs. Select a user group to the right in the dialog, and press the **Add** button to make the user a member of other user groups.
10. Select a group to the left and click on remove to remove a group from the user. Apply the changes with the **Apply** button.



Aspects of 'AndyC'	Modified	Descr...	Inherited
User Structure	10/19/99 15:01:36		False
Admin Structure	10/19/99 14:38:53		False
Name	10/19/99 15:00:44		False
Object Icon	02/28/99 21:00:06		True
Process Dialog Profile Values	05/04/99 12:27:26		True
Trend Profile Values	06/01/99 17:26:48		True
Type Name	10/12/99 15:14:51		True
User Definition	10/19/99 15:01:36		False
User Structure	10/19/99 14:38:53		False
User Structure	10/19/99 14:38:53		False

Advant User Account: AndyC

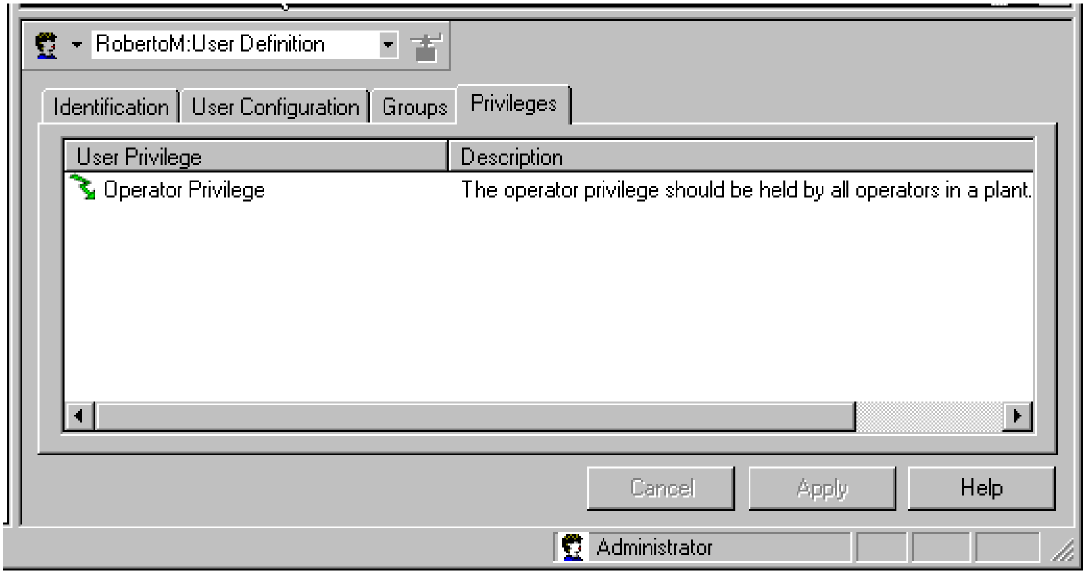
Member of:	Not member of:
Everyone	Software Developers
System Engineers	Operators
Application Engineers	

Figure 19-9. Assigning user groups

11. User Privileges are assigned when a user is added to an Advant User Group. User Privileges provide a basic authority control in the Advant system and adapts your environment to the type of user you are. The following User Privileges exist:

  - Application Engineer
  - Operator
  - System Engineer
  - Software Developer

12. User privileges help customize the environment to suit the needs of each user. You can use the Privileges tab of the User Definition aspect to show the privileges a user holds. The Privileges Tab is for viewing only. The result from the steps above could look like this for the Operator Roberto Mancini.


RobertoM:User Definition

User Privilege	Description
Operator Privilege	The operator privilege should be held by all operators in a plant.

Figure 19-10. Roberto Mancinis privileges

Chapter 20 Building History and Trend Applications

20.1 Building History Applications

Most of the time, the History applications are built very late in the application work. The Signal objects are available and you know exactly how you want to log most of the Signals. It is recommended that you follow these steps:

The different steps are described in detail in chapter 3 of *Advant History User's Guide*.

1. Start by grouping the signal objects into groups which you want to log in the same way.
2. For each group define the log attributes for the primary and for any secondary logs you want. Here is an example on how it may look like:

Table 20-1. Example group for all Silo Levels

Log level	Sample interval	Log Period	Start Time	Storage interval	Bad data Quality limit	Calculation Alg
Primary	15s	4 hours	00:00:00:0 1-jan 1999	30s	0	Instantaneous
Secondary 1	1m	12 hours	-”-		10%	Average
Secondary 2	5m	48 hours	-”-		10%	Average

We are assuming for this group that we are not using deadband compression. We assume the defaults for AdvaSoft 2.0 for Windows NT for the collection mode, which are synchronous, collection type which is OPC, storage type which is Type-1.

3. Use the History builder to build a log aspect according to your group table for one of the objects that will be logged according to this. Repeat for each one of the groups. These log aspects shall now be used as log aspect templates.
4. Run the dbConfig utility to dump those log aspect templates into a text file.
5. Use a text editor or any other available tool to create a tag list for each one of the group of objects to log. The tag list should contain both the object name and the attribute name, embedded with any delimiter necessary (in this case the ‘:’)[1]. Example: AC110_11_AIS9:VALUE.
6. Use dbConfig to create a bulk configuration file with the “like” syntax using the template file and a tag list for each one of the objects to log.

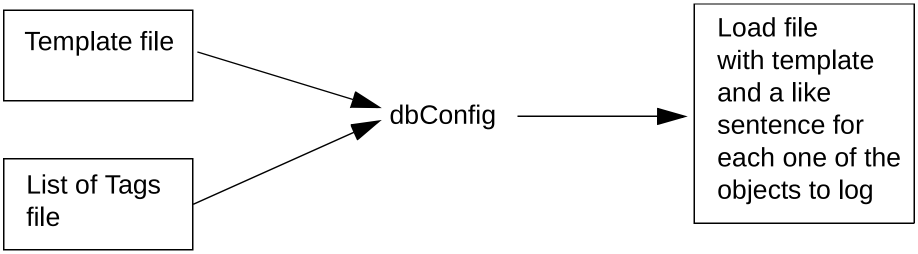

Template file
List of Tags file
dbConfig
Load file with template and a like sentence for each one of the objects to log

Figure 20-1. Creating a file with the like syntax

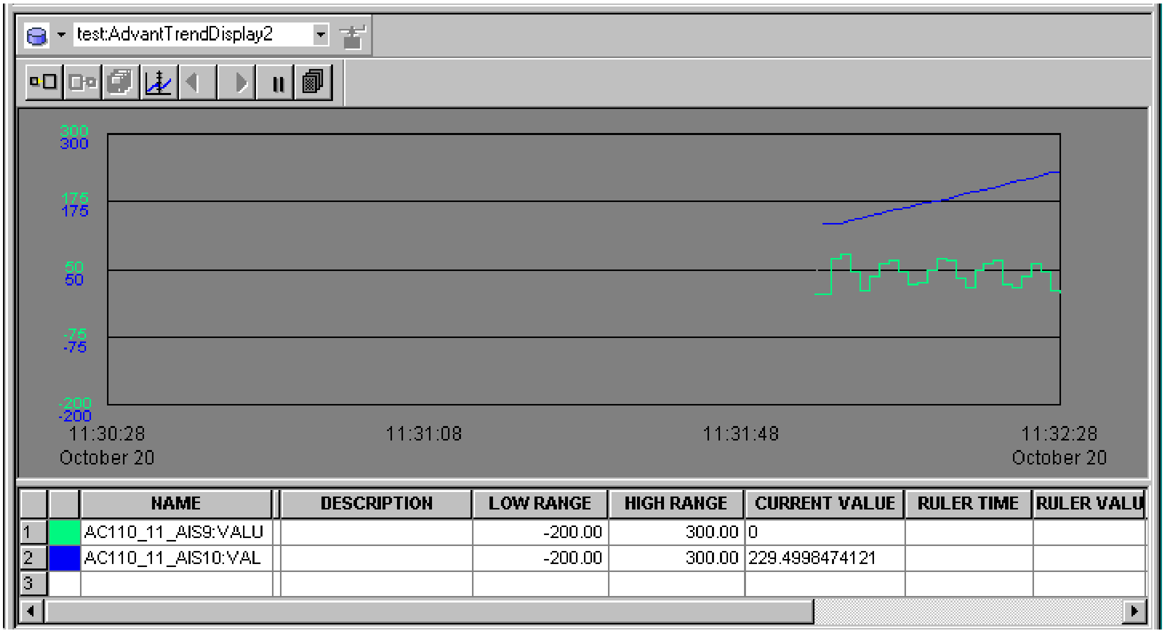
7. Use dbConfig to parse and check all the resulting text files.
8. Use dbConfig to load the complete History database from the text files. dbConfig checks the syntax of the loaded files and verifies the configuration information. If there are no errors, dbConfig creates the history objects defined in the files for the applicable node.
9. This is a good time to backup your system before continuing. You need to restart your system anyhow to get the new History configuration activated, the bulk configuration is an off-line activity.

---

[1] Different OPC servers may have different separators.

20.2 Building Trend Applications

As a default, each signal object will have a Trend aspect. It will present an on-line log of the current object value if no History log exists. If a History log exists the Trend presentation will read and present the History log and then append an on-line logging of the current value.



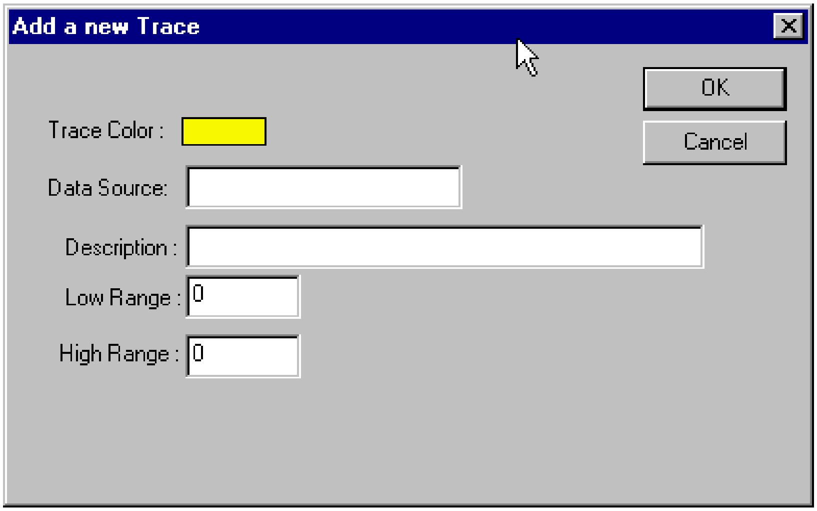
		NAME	DESCRIPTION	LOW RANGE	HIGH RANGE	CURRENT VALUE	RULER TIME	RULER VALU
1	🟩	AC110_11_AIS9:VALU		-200.00	300.00	0		
2	🟦	AC110_11_AIS10:VAL		-200.00	300.00	229.4998474121		
3								

Figure 20-2. Trend showing On-line logging

Going back to our Milko Chemical plant. We want a trend aspect on each of the process cells as well. It will present 6 of the most essential signal objects over time. To build it is very easy:

1. Create a Trend aspect for the Process Cell object. (You should know how to create an aspect by now).

2. To add a trace, choose **Insert Trace** from the right-click menu, and then use the Add New Trace dialog, Figure 20-3, to configure the following attributes Trace Color, Data source (Object Name: Attribute name, Description, Low Range, High Range.



Trace Color : 🟨
Data Source:
Description :
Low Range : 0
High Range : 0

Figure 20-3. Adding a trace to a trend display

3. Repeat step 2 for all traces (a maximum of 6) that you want in the trend aspect.

Chapter 21 Building Complementary Applications

AdvaSoft 2.0 for Windows NT has a number of aspects that can be used to access various software tools and the information they provide:

- The Web aspect
- The ActiveX Wrapper aspect
- The Windows application aspect

The Web aspect gives you possibility to access all information published as HTML pages. Everything published on your intranet (and if you allow access to the internet) can be accessed. Datasheets and local intranet information related to certain functionality are typical things to include. In our example we want to connect to the suppliers to some of our devices. (Pumps and Valves). See Section 21.1, Adding a Web Aspect showing Technical Data for a Valve.

ABB does not recommend that you configure links outside your own Intranet without commitments from your IT manager/responsible.

The ActiveX Wrapper aspect makes it possible to wrap any ActiveX Control. There are a number of third party and ABB applications including ActiveX Controls that may be of interest to integrate into your Advant Workplace products. The ActiveX Wrapper provides such a feature. See how we wrap an ActiveX in Section 21.2, Add an ActiveX Wrapper Aspect.

The Windows application aspect makes it possible for you to launch most types of Microsoft or third party tools: Wordpad, Notepad, MS-Word, Excel etc. This makes it possible to view or to update various types of documents and information. As you have seen before in this document we use MS-Word as document editor. In Section 21.3, Adding a Windows Application Aspect opening a Word document you will see an example on how to do that.

21.1 Adding a Web Aspect showing Technical Data for a Valve

Creating a Web Aspect showing the Technical Data of the V40 valve

An example how to create a Web Aspect towards the Valve suppliers Internet pages. Follow the steps below:

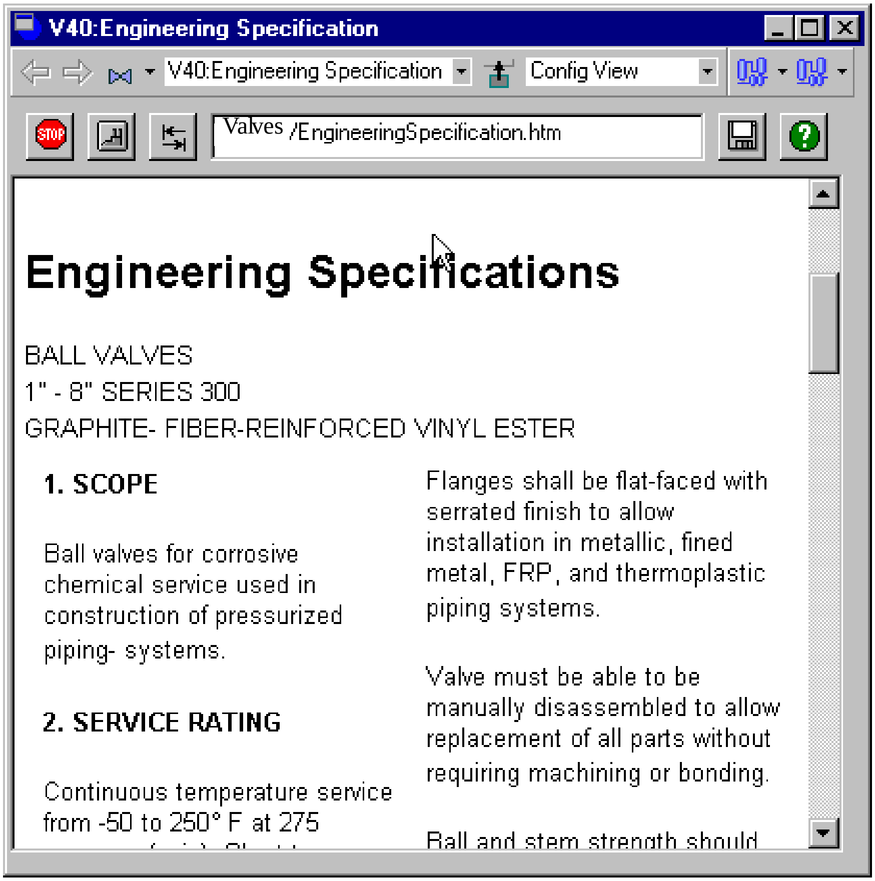
1. Open the Advant Explorer and go to the Process Object V40, representing the Valve.
2. Select the Object and create a “Web Page” aspect.
3. Right click on the new aspect. Choose “Config” in the Context Menu that opens.

   If you click on the **Config View** line, you open a configuration window for the Advant Browser.

If you click on the Explorer line in the context menu, the MS Internet Explorer opens. How to configure and use this browser, please see the relevant Microsoft Explorer documentation.

4. You choose to use the Advant browser. Next step is to type the address to the Valve page in the WWW address field.
5. When you have typed the right address, click on the SAVE button.

6. The configuration is done. Bring the aspect up to see that it works.


V40:Engineering Specification
V40:Engineering Specification
Config View
STOP
Valves /EngineeringSpecification.htm

Engineering Specifications

BALL VALVES
1" - 8" SERIES 300
GRAPHITE- FIBER-REINFORCED VINYL ESTER

1. SCOPE

Ball valves for corrosive chemical service used in construction of pressurized piping- systems.

2. SERVICE RATING

Continuous temperature service from -50 to 250° F at 275

Flanges shall be flat-faced with serrated finish to allow installation in metallic, fined metal, FRP, and thermoplastic piping systems.

Valve must be able to be manually disassembled to allow replacement of all parts without requiring machining or bonding.

Ball and stem strength should

Figure 21-1. Web page for our Valve

For more details see the *Advant Workplace User's Guide*.

21.2 Add an ActiveX Wrapper Aspect

The ActiveX Wrapper provides you with a method to wrap any ActiveX Control. As a result you can use ActiveX components from various suppliers but you can also use some of the ActiveX Controls that are available as part of Windows NT such as the calendar. When you have configured the ActiveX Control you simply select that aspect to execute it.

Creating an ActiveX Wrapper Aspect

An example how to create an ActiveX Wrapper Aspect. Follow the steps below:

1. Open Advant Explorer and go to the Object, to which you want to add an ActiveX Wrapper Aspect.
2. Select the Object and create an “ActiveX Wrapper” aspect.
3. Right click on the new aspect. Choose “Config” in the Context Menu that opens.
4. A dialog box opens. Click on the right tab.
5. Type in the path to the ActiveX Control, that you want to access, in the input field or use the Browser.
6. When you have typed in the path or used the browser, click on the **OK** button, and the Display dialog box closes and you see the ActiveX Wrapper dialog box again.
7. If you click on the **Preview** button, you will see a preview of the ActiveX Control.
8. To set the ActiveX properties click on the **Properties** button.

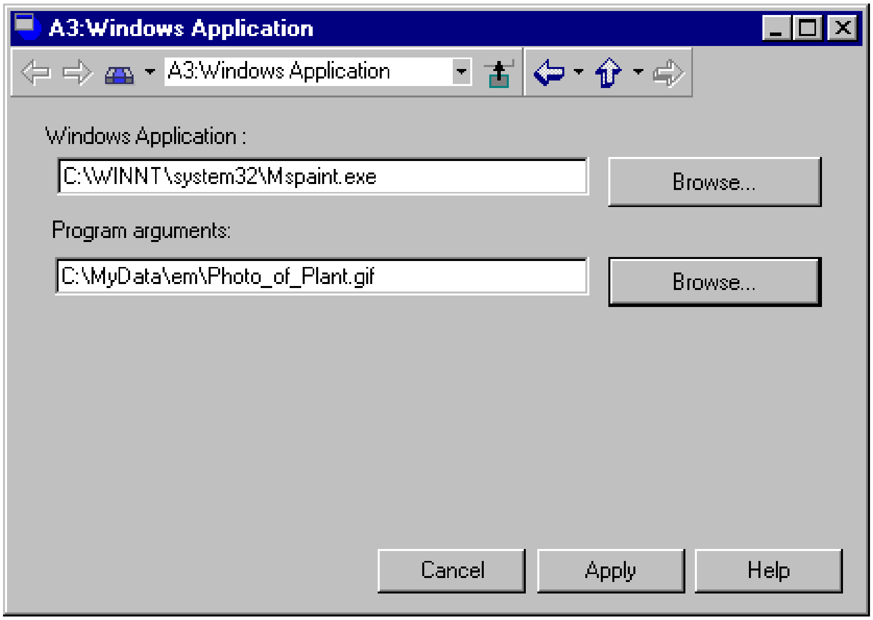
For more details see the *Advant Workplace User’s Guide*.

21.3 Adding a Windows Application Aspect opening a Word document

Creating a Windows Application Aspect to execute MS-Word

Follow the steps below:

1. Open Advant Plant Explorer and go to the Object, to which you want to add a Windows Application Aspect.
2. Select the Object and create a “Windows Application” aspect.
3. Right click on the new aspect. Choose “Config” in the Context Menu that opens.
4. A dialog box opens. Click on the right tab. See Figure 21-2.



A3:Windows Application

Windows Application : C:\WINNT\system32\Mspaint.exe

Program arguments: C:\MyData\em\Photo_of_Plant.gif

Figure 21-2. Windows Application Aspect Context Menu

5. Browse to the path of the Windows application and click OK

In a multi-client system, make sure you point out the executable in such a way that it can be found from each one of your Advant Workplaces.

6. Type in the File path of the file that you want the application to open. In our case this is the path to a graphical file.

In a multi-client system, make sure you point out the file to use in such a way that it can be found from each one of your Advant Workplaces. That is, you need to place it in a shared directory.

For more details see the *Advant Workplace User's Guide*.

Chapter 22 Backup of Application Data

To secure the Advant Workplace Products installations and the applications that you build, the following backup strategy is recommended, see also *Advant Workplace Products Administrators Guide* for detailed instructions:

- When you have installed all the products on the PC:s and verified that they are up and running you should secure them with a full backup. Our recommendation is to use the software GHOST. See also www.symantec.com for more information
- To secure application data you can use three alternative methods:
  - Make a full backup of your server disk(s) using Ghost or similar software, whenever it is convenient to shut down the system
  - Use NT backup, to make off-line backups of application data.
This includes all application data including the History data.
  - Use the Export and Import feature to make an on-line backup of application data. A full export includes all application data except the History configurations and logs.

Appendix A Example on Reference Material

A.1 Milko Milkshakes

A.1.1 Process Description

1. Additive deliveries
2. Additive Warehouse
3. Solid Processings
4. Intermediate Product Store
5. Milk Reception
6. CIP Chemical Deliveries
7. CIP Chemical Store
8. CIP Generation System
9. Liquid Process Buildings
10. Packaging Building
11. Filling
12. Quarantine Store
13. Final Product Store

A.2 Overview

A.2.1 Products

- Splodge Drinks: Banana, Strawberry and other fruit based drinks.
- Fudge Drinks: Chocolate and other cocoa based drinks.
- Alco shakes: Milk shake, alcoholic drinks.

A.2.2 Recipes

Not just different products but also different packaging and formulas.

- Premium Supermarkets
- Milkos own brand
- Lower price supermarkets

A.3 Process Cells versus Common Resources

3 Process Cells

- Solid processing cells
- Milkshake cell
- Packaging cell

10 Common Resources

- CIP (+CIP Chemical store)
- Additive warehouses
- Control room
- Laboratory
- Final Product/Intermediate/Packaging/Quarantine store.
- WWTP

A.4 Milkshake Plant Process Descriptions

Additive Deliveries

On arrival of a lorry at the Additives Reception bay the pallets of additives are unloaded by forklift truck and taken to the Additive Warehouse.

A procedure is required to register the arrival, which should match an expected delivery as scheduled in the MES.

Additive Warehouse

The Additive warehouse stores palettes of flavor additives. Each pallet contains a number of sacks or additives. A system to control the location of the materials and to ensure that they are used in an appropriate order, such as first-in first-out according to use by date will be required. This document does not explore these areas of functionality.

Solid Processings

Flavor additive materials are depalletised then the sacks are slit the contents extracted and then pneumatically conveyed into the Additive Silo. Additives are then milled, sieved and then weighed into IBC's. This is driven by the production campaign requirements so that the right additives in the right quantities are processed.

Intermediate Product Store

The Intermediate Product Store provides a buffer which is used to store IBC's which have been prepared in advance of when they are needed or in case of a hold up in the liquids processing.

Milk Reception

On arrival of a tanker at the mile reception the tanker is connected up to on e of the three possible unloading points. Milk is then pumped from the tanker into one of the three milk silos. On completion of the unloading of the tanker a clean is carried out of the tanker and the pipework which routes the milk from the tanker to the silos. The milk silos operate on a continuous cycle whereby they are filled repeatedly during the day and then cleaned at night, when no milk deliveries are expected. Only one tanker is unloaded at one time and generally to only one milk silo. However, operators can in exceptions unload into one silo until it is full and then the next.

CIP Chemical deliveries

The CIP Chemical Store stores drums of cleaning chemicals. A system to control the location of the drums and to ensure that they are used in an appropriate order, such as first-in, first-out or according to use by dates will be required. This document does not explore these areas of functionality.

CIP Generation System

The Clean in Place system provides rinse water and also doses the cleaning chemicals into the water to make cleaning liquids. The system also takes the returned liquids and can store the final rinse water for use in subsequent first rinses. A more detailed description is given below.

Liquids Processing Buildings

Batch Mixing Processes are carried out to make batches of the various products. Certain products are then homogenized on their way to the Product Silos.

A typical mixing process comprises:

- Add Milk Start qty
- Add Milk and Additive
- Add Final Milk qty

Mix

- Mix and Heat
- Transfer Out

Packaging Building

In the packaging building liquid product from one of the product silos is filled into bottles and then labeled and packed into pallets - further details are given below. The operation of the packaging line is organized in terms of Work Orders.

A Work Order is characterized by:

- The number of bottles to be packed.
- The labeling on the bottle - one work order per label type.
- The customer - a single customer per work order.
- The single work order may be made up from one or more product silos.

Filling

Bottles are supplied to the filling line in boxes of pallets. A depalletising machine takes the boxes from each pallet and the de-boxer takes bottles from the boxes. Boxes are then fed onto an accumulation conveyor, which feeds into the filling machine. Bottles are filled, capped, labeled placed in boxes and then the boxes are shrink wrapped and then palletized.

Quarantine Store

Product is held in the product store until it is requested by a delivery work order.

Customers

Milko Ltd supplies to a large number of customs including the premium supermarkets, Milko Ltd’s own brand and some others. Milko also has a strategy to be able to supply new customers in response to order placed over the phone or the Internet when the customers usual supplier is unable to meet the demand or has a production problem. The pricing of such orders is a strategic decision made by Milko’s senior management - sometimes the cost will be high in order to make sure that the cost of the disruption to the schedule are more than covered. However, sometimes the work is done at cost or even at a loss if the new customer is considered strategically important enough.

Rework

Just occasionally, and notwithstanding the highly effective scheduling system, a batch is made which does not meet the critical quality standards of Milko Ltd. This can occur where there is an equipment failure - in general equipment failures can be tolerated by putting the process onto hold, repairing the failed item, and then re-starting. However, sometimes there are failures, which cannot be fixed in time to allow the batch to be completed. In these cases it is highly preferable to be able to re-use the batch if possible. This is far more acceptable than disposing of the batch down the effluent system, and can be achieved by using small percentages in the plant makes it possible to divert a failed batch back into one of the milk silos for subsequent rework.

A further issue with rework is that when a problem does occur and some rework is needed, it can become necessary to change the production schedule to make products, which can use up the reclaimed batch. (It may also require that the current schedule is altered in order to free space in a Milk Silo to hold the batch).

Recipes

The plant produces a large number of products, in a variety of packages. There are three basic types of recipes.

Splodge Drinks

Recipe characteristic

Each type of product is made in a different way.

Within each general recipe type there are a large number of specific recipes. The Premium Supermarkets and Milko themselves each have their own recipes but the lower price customers all use exactly the same recipe. Rework is handled by a rework substitution algorithm - this is a set of recipes par4ameter which define how much of a particular type of rework batch can be added to all other products which also contain banana, no rework which contains alcohol can be used in any non-alcoholic products etc.

Packaging

The product is packed in a variety of ways. Each customer has their own labels for the bottles or cartons, which are used. Alcoshake is provided in bottles with child proof caps and each customer has their own bar-coding standard.

Operations

Homogenization

Homogenization is essentially mixing process, which ensures that the various constituents of a batch are thoroughly mixed. The depth of a mix is determined by the physical properties of the batch and the pressure and flow through the homogenizes.

Homogenization, if required takes place in the process of transferring a batch from a Batch vessel to a Product Silos.

The process starting up a homogenize is complicated by the mechanical problems of raising the pressure which can be very high and the fact is that is possible to physically damage the equipment by running it empty. The homogenizer supplier provides a control system specifically for this equipment and refuses to guarantee it if this system is not used. Consequently Milko has bought the homogenizer and its control system has a complete package. The vendors system includes a serial interface so that all of the control parameters, which the vendor considers relevant, can be set from a supervising system (There are further problems, which result from this).

A.5 CIP

A.5.1 Introduction

The CIP Station for the milk shake plant essentially consists of one fresh water tank, one Return Water Tank, one Hot Caustic Tank and two feed and return lines. The CIP station is used to clean the Milk Silos area, the Liquid Processing area and the Product Silos. These are two types of CIP sequences, line CIP and tank CIP.

The typical CIP station has two feed and return lines. The CIP Station on the Milkshake Plant might require additional feed and return lines or routing valves to distribute the CIP liquids to the several CIP feed and return lines shown on the Milkshake P & ID.

A.5.2 CIP Process Description

Line CIP

The normal sequence for a line CIP is as follows:

1. Pre-rinse	Return water tank to drain at CIP station via selected line.
2. Push out water with caustic	Caustic tank to drain at CIP station via selected line.
3. Caustic wash	Caustic tank to caustic tank via selected line.
4. Push out caustic with water	Fresh water tank to Caustic tank via selected line.
5. Post caustic wash	Fresh water tank to return water tank via selected line.
6. Push out water with acid	Acid tank to return water tank via selected line.
7. Acid wash	Acid tank to acid tank via selected line.
8. Push out acid with water	Fresh water tank to acid tank via selected line.
9. Final rinse	Fresh water tank to drain via selected line.

Tank CIP

The normal sequence for a tank CIP is as follows:

1. Pre-rinse	Return water tank to drain at CIP station via selected line.
2. Empty tank	Empty tank to drain at CIP station via selected line.
3. Push out water with caustic	Caustic tank to drain at CIP station via selected line.
4. Caustic wash	Caustic tank to caustic tank via selected line.

5. Empty tank	Empty tank to caustic tank via selected line.
6. Push out caustic with water	Fresh water tank to caustic tank via selected line.
7. Post caustic wash	Fresh water tank to return water tank via selected line.
8. Empty tank	Empty tank to return water tank via selected line.
9. Push out water with acid	Acid tank to return water tank via selected line.
10. Acid wash	Acid tank to acid tank via selected line.
11. Empty tank	Empty tank to acid tank via selected line.
12. Push out acid with water	Fresh water tank to acid tank via selected line.
13. Final rinse	Fresh water tank to drain via selected line.
14. Empty tank	Empty tank to drain via selected line.

The line and tank CIP’s in the Milkshake plant would usually only include the acid wash once a week or once a fortnight. If the acid wash is not included then steps 7 through 12 are skipped.

The sequence for cleaning the Batch mixing tanks will probably include some additional steps. During wash steps it would also be necessary to pulse sample valves etc.

Assuming that there are two feed and return lines then two CIP’s can operate in parallel. This means that one of the mix tanks could be cleaned at the same time as one of the tanker intake lines and that the liquid from one of the CIP tanks (for example acid) could be circulated through the intake and the mix tank at that time.

In the post caustic wash fresh water is recovered to the return water tank. This leaves the possibility for cross contamination.

A.5.3 CIP tank Make-up

The normal sequence for a CIP tank Make-up is as follows:

1. Dump tank contents	Transfer contents of CIP tank to effluent tank.
2. Fill tank with water	Fill tank to high level with fresh water.
3. Dose chemical concentrate	Empty tank to drain at CIP station via selected line.
4. Heat tank	Caustic tank to drain at CIP station via selected line.

Fresh water tank requires only step 1.

Cold caustic and acid tank only require step 1 and 2.

Hot caustic tank requires steps 1, 2 and 3

A.5.4 CIP Recipe

The recipe for a CIP will usually consist only of a set of times for each step and a single on/off bit to decide whether to include an acid wash in the CIP sequence.

It is possible to include additional items such as pre-set values for watch dog timers for push out and emptying steps etc. to allow the operator to tune the CIP’s. It would also be possible to allow the sequence of steps to be modified as part of the recipe but this is not really necessary.

Therefore a CIP recipe can essentially be reduced to the following:

1. With Acid Y/N	1/0
2. Pre-rinse	10 minutes
3. Caustic wash	20 minutes
4. Post caustic wash	8 minutes
5. Acid wash	15 minutes
6. Final rinse	15 minutes

A.6 S88 Analysis

The following table lists the main S88 entities:

Table A-1. Main S88 entities

Drawing object	S88 Definition	S88
Additive Warehouse	Common Resource	Additive Warehouse
CIP Chemical Store	Common Resource	CIP Chemical Store
CIP Generation System	Common Resource	CIP Generation System
Control Room	Common Resource	Control Room
Final Product Store	Common Resource	Final Product Store
Intermediate Store	Common Resource	Intermediate Store
Laboratory	Common Resource	Laboratory (Maybe not a S88 object really)!
Packaging Store	Common Resource	Packaging Store
Quarantine Store	Common Resource	Quarantine Store
Water Treatment Plant	Common Resource	Water Treatment Plant
Milk Silos	Process Cell	Milkshake Cell
Liquids Processing Buildings	Process Cell	Milkshake Cell
Product Silos	Process Cell	Milkshake Cell
Packaging Building	Process Cell	Packaging Cell
Solids Processing Building	Process Cell	Solids Processing Cell

We have identified the Process Cells and the top level Common Resource S88 objects in the plant.

As this study is concerned with fitting in those parts of a real plant that are not well defined by S88 we have taken the liberty to call them Common Resources.

These are all parts of the same process cell.

Explanation

There are 3 process cells and 10 Common Resources at the Plant Level. See below for discussion.

Process Cells

In this analysis we have enforced a rule that one batch = one process cell (although there may be several in progress), so making it necessary to split the plant down onto several process cells.

Thus:

The Solids Processing Cells delivers batches to IBCs.
The Milkshake Cell delivers batches to product silos.
The Packaging Cell delivers batches to the Quarantine Store.

Common Resources

We have as mentioned identified 10 common reserves at the plant level. That is to say that these are common resources that are hierarchically directly below the S88 Plant Area Cell and do not even appear to belong to a Process Cell. This is feasible because they provide services to several process cells and therefore cannot be logically subordinated to them.

- The additive warehouse serves the solids processing and so might be placed inside that cell.
- The CIP Chemical Store serves the CIP Generation System and so might be placed that cell.
- The CIP Generation System serves the Milkshake cell and the Packaging Cell.
- The Control Room is not a service to the batch but is certainly involved in the batch control.

- The Laboratory may be carrying out operations on sample of the batch but is not processing.
- The Final Product Store, Intermediate Store, Packaging Store and Quarantine Stire are all providing a service to the plant.
- The Water Treatment Plant serves the entire plant.

It is debatable that these are all really common resources according to S88, however, we need to put all of the equipment in the plant into the physical model in some way.

The next stage of the analysis involves defining in more detail the objects we have one listed.

- Process Cells
- Solids Process Cell

The following table lists the main S88 entities:

Table A-2. S88 Drawing objects

Drawing object	S88 Definition	S88
Solids Prep Line 1	Unit	Solids Prep Line
Solids Prep Line 2	Unit	Solids Prep Line
Blending	Unit	Blending

Liquids Processing Cell

There are three parts which make up the Liquids Processing Process Cells. Milk Reception, Liquid Processing Building and Product Silos:

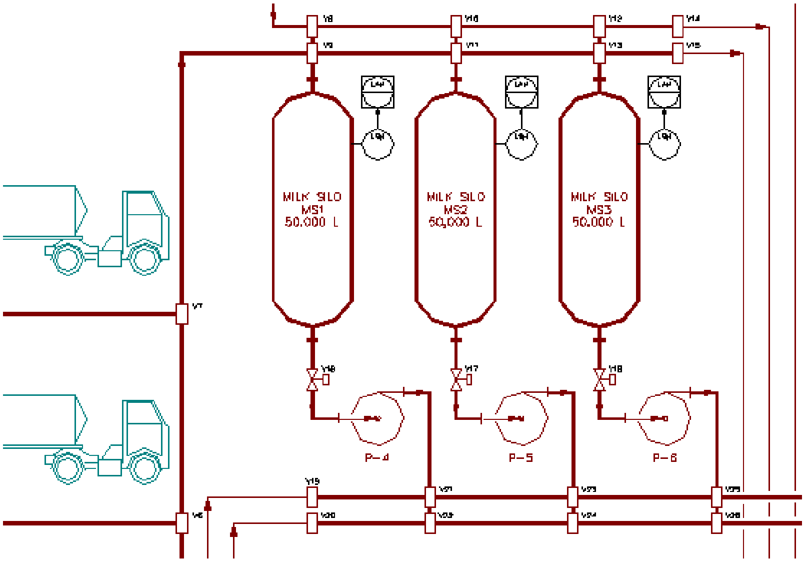


Figure A-1. Milk Reception Silos

Please excuse the bad quality on the figures containing drawings. They were scanned in from very wrinkled originals.

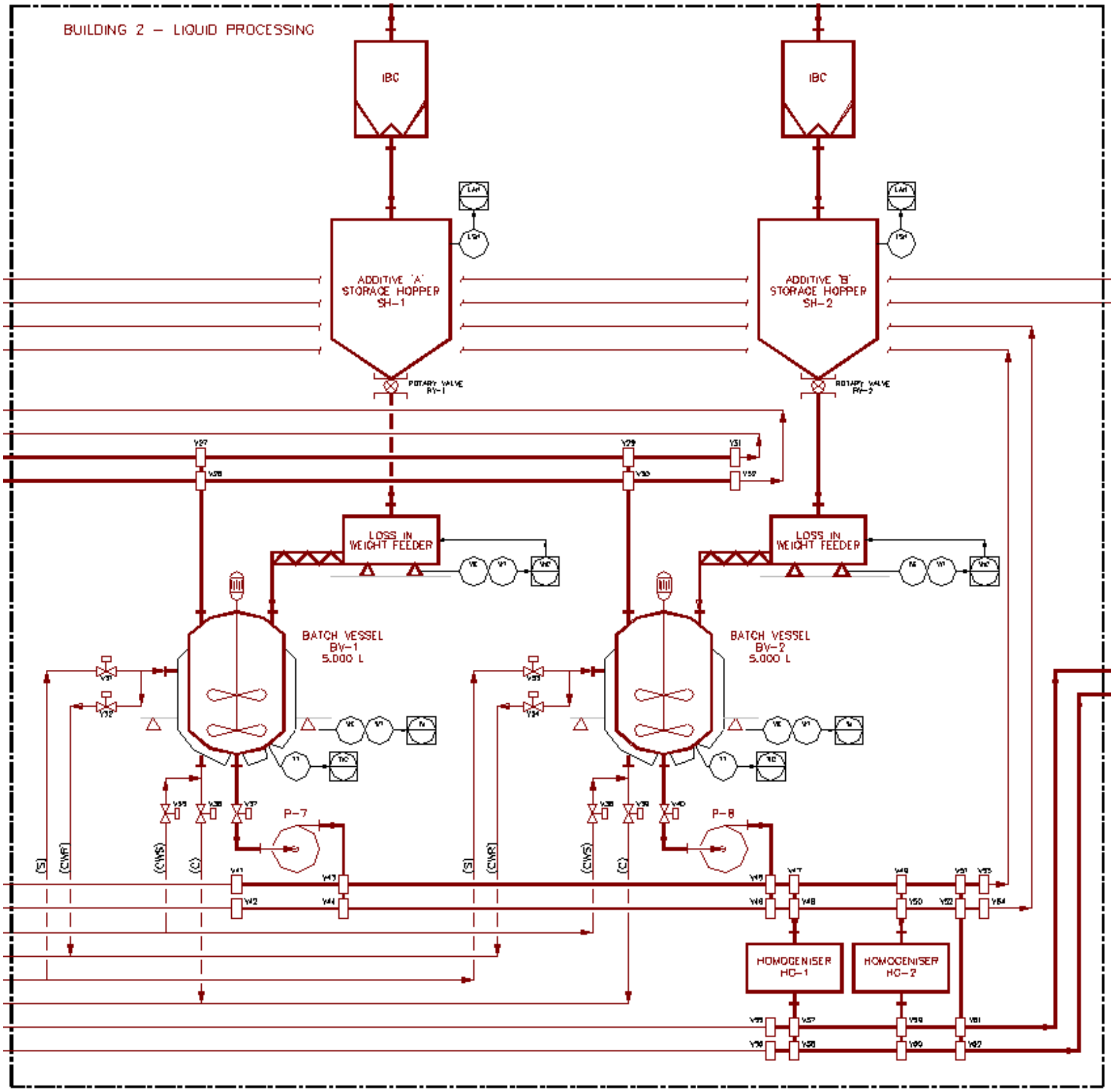


Figure A-2. Liquids Processing Building

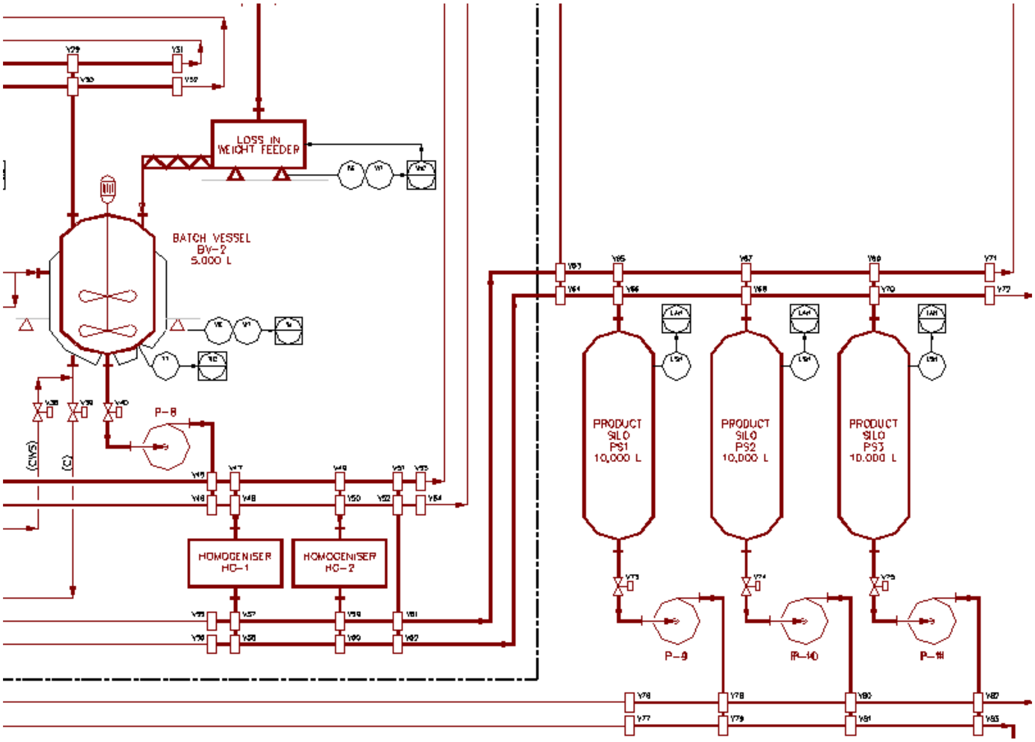


Figure A-3. Product Silos

The following table lists the main S88 entities in the Liquid Processing Cell:

Table A-3. S88 entities in Liquid Processing Cell

Drawing object	S88 Definition	S88
EM1	Common Resource	Distribution pipework
EM2	Common Resource	Distribution pipework
EM3	Common Resource	Distribution pipework
EM4	Common Resource	Distribution pipework

Table A-3. S88 entities in Liquid Processing Cell (Continued)

Drawing object	S88 Definition	S88
EM5	Common Resource	Distribution pipework
HG2	Equipment Module	Homogeniser
HG1	Equipment Module	Homogeniser
EM5	Equipment Module	Tanker Unloading Bay
EM5	Equipment Module	Tanker Unloading Bay
EM3	Equipment Module	Tanker Unloading Bay
EP_Receive_milk	Equipment Procedure	Receive Milk
CIP_L1	Equipment Procedure	Line CIP
CIP_L2	Equipment Procedure	Line CIP
CIP_L3	Equipment Procedure	Line CIP
CIP_H1	Equipment Procedure	Line CIP
CIP_L4	Equipment Procedure	Line CIP
BV1	Unit	Batch Mixing Unit
BV2	Unit	Batch Mixing Unit
MS1	Unit	Milk Silo
MS2	Unit	Milk Silo
MS3	Unit	Milk Silo
PS1	Unit	Product Silo
PS2	Unit	Product Silo
PS3	Unit	Product Silo
SH1	Unit	Additive Storage Hopper
SH2	Unit	Additive Storage Hopper

Common Resources in Process Cells

These distribution manifolds are examples of Share resources with the possibility of multiple users:

- Milk Supply manifold
- Milk Distribution Manifold
- Product Homogenizing manifold
- Product Distribution Manifold
- Units
  - Solids Preparation Line
  - Blending
  - Additive Storage Hopper
  - Milk and Product Silos
  - Batch Mixing Unit
  - Packaging Unit
- Other Equipment Modules
  - Tanker Unloading Bay
  - Homogenizer
- Equipment Procedures
  - Line CIP
  - Vessel CIP
  - Receive Milk
  - Recipe Procedures
  - Typical Recipe.

A.7 Object Count

Table A-4. Signal list

Class	Type	Count
Alarm	AH	9
	AHH	9
	AL	9
	Alarm	102
	ALL	9
	FTC	107
	FTO	107
	FTR	39
	FTS	39
	PAHH	2
	PAL	2
	PLC Warning	6
	PLC Fail	16
	PVHi	2
	PVHiHi	2
	PVLo	2
	PVLoLo	2
	XA	12
	Total	***476***

Table A-4. Signal list (Continued)

Class	Type	Count
Common Resource	Additive Warehouse	1
	CIP Chemical Store	1
	CIP Generation System	1
	Common Resource	6
	Control Room	1
	Distribution pipework	5
	Final Product Store	1
	Intermediate Store	1
	Laboratory	1
	Packaging Store	1
	Quarantine Store	1
	Water Treatment Plant	1
	Total	***21***
Control Module	Accumulation Conveyor	2
	AG	5
	CI	1
	CIP_Check	1
	Conveyor	4
	Fan	2
	FI	2
	LAH	24
	LAL	21
	Line Control	1

Table A-4. Signal list (Continued)

Class	Type	Count
	Machine Modes	14
	Metal Detector	2
	Motor CM	3
	Packing Conveyor	2
	Pump	18
	RV	7
	SIA	2
	TC	2
	TI	3
	TIC	1
	Unit control Module	2
	Weight Amp Control	10
	WIA	6
	WQ	2
	WT	4
	XV	110
	Total	***251***
Control System	AI	15
	Alarm Event Printer	1
	AO	3
	BCP	6
	BCR	5
	Computer	1

Table A-4. Signal list (Continued)

Class	Type	Count
	Controller	10
	DCS Workstation	3
	DI	313
	DO	146
	Equipment Controller	18
	Local Display	5
	MES Workstation	6
	MIS Computer	1
	PCS Controller	6
	PCS Server	1
	Printer	3
	Report Printer	1
	SI	28
	Workstation	4
	Total	***576***
Datastore	Analog value	16
	Binary Value	510
	Total	***526***
Effector Element	CV	1
	Motor Starter	39
	Sov	107
	Total	***147***
Equipment Module	EM	22

Table A-4. Signal list (Continued)

Class	Type	Count
	Homogeniser	2
	Hopper	8
	Tanker Unloading bay	3
	Total	***35***
Equipment Procedure	Collect Pallet	5
	EP	10
	Line CIP	5
	Receive Milk	1
	Receive Pallet	5
	Total	***26***
Logic Function	And Gate	3
	Difference	1
	Motor Driver	39
	Or Gate	0
	Select	2
	User Maths	1
	Valve Driver	107
	Total	***153***
Loop Function	_C	2
	Characterise	2
	Compare	0
	Total	***4***
Measuring Element	FT	2

Table A-4. Signal list (Continued)

Class	**Type**	**Count**
	Load Cell	10
	LSH	18
	LSL	21
	Mrun	39
	QT	1
	TT	6
	ZSC	107
	ZSO	113
	Total	***317***
Operation	Manual Operation	10
	Operation	0
	Sample Operation	2
	UO	14
	Total	***26***
Operation Display	Indicate	167
	Total	***167***
Operator Entry	HS	508
	SP	305
	Total	***813***
Phase	Phase	53
	Total	***53***

Table A-4. Signal list (Continued)

Class	Type	Count
Process Cell	Milkshake Cell	3
	Packaging Cell	1
	Solids Processing Cell	1
	Total	***5***
Recipe	RCP	6
	Total	***6***
Unit	Additive Storage Hopper	2
	Batch Mixing Unit	2
	Blending	1
	Milk Silo	3
	Packaging Line	1
	Product Silo	3
	Solids Prep Line	2
	Unit	1
	Total	***15***
Unit Procedure	UP	17
	Total	***17***
Total number of instances		***3634***

Appendix B Advant Structures

B.1 Advant Structures

This section gives an overview of the basic structures that are present in Advant Plant Explorer when the system has been installed. The system comes with twelve predefined structures. The set of structures may be extended by an arbitrary number of structures by anyone using the Advant system. The structures described have, in some cases, been filled with more data than when first installed.

The structures described in this section are needed for correct system operation and must not be removed or modified in any substantial way. Normally, these structures enforce their own rules, for example what can be inserted.

Each structure is described in more detail in the sections below. The Advant system consists of three primary structures:

- Functional Structure
- Location Structure
- Control Structure.

These structures are primary structures because it is assumed that most Operators and Application Engineers will perform most of their work within these structures.

Some additional and important structures are the:

- Admin Structure
- Object Type Structure
- Aspect System Structure.

These structures contain the objects that are used to configure and control the system itself.

The administration structures are all structures that are pre-created and which may contain only objects that are placed in the system domain. These objects typically have restricted access, requiring System or Application Engineer rights to allow modifications. A set of additional administration structures exists. These structures all have different purposes and they control much of the Advant system behavior.

The addition administrative structures are:

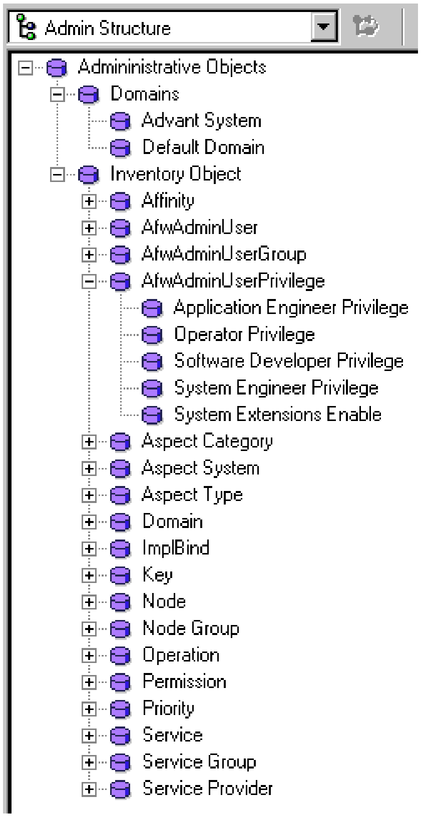
- The Service Structure which holds service definitions.
- The Workplace Structure which contains the workplace definitions of a system.
- The User Structure in which all user groups and users are placed.
- The Node Administration Structure which holds all node groups (sometimes called networks) and nodes defined in the system.
- The Graphics Structure which defines graphic primitives and subelements.

Each structure is implemented as an aspect category and has therefore its own authority control etc. Every structure with objects has at least one root object. A structure may have multiple root objects, and these are displayed as top-level objects in Advant Plant Explorer.

B.1.1 Admin Structure

The system contains several administrative structures. One structure exists which is called the *Admin Structure*. This structure contains an inventory of all admin objects in the system placed under an object called *Inventory*.

The *Admin Structure* uses the Administrative Objects root object as its root. It contains two objects, called *Domains* and *Inventory.* The *Inventory* object contains all aspect type objects of the aspect types that implement an admin object as its child. This means that all aspect systems, aspect types, affinities etc. are easily found in one place. See Figure B-1.



Admin Structure

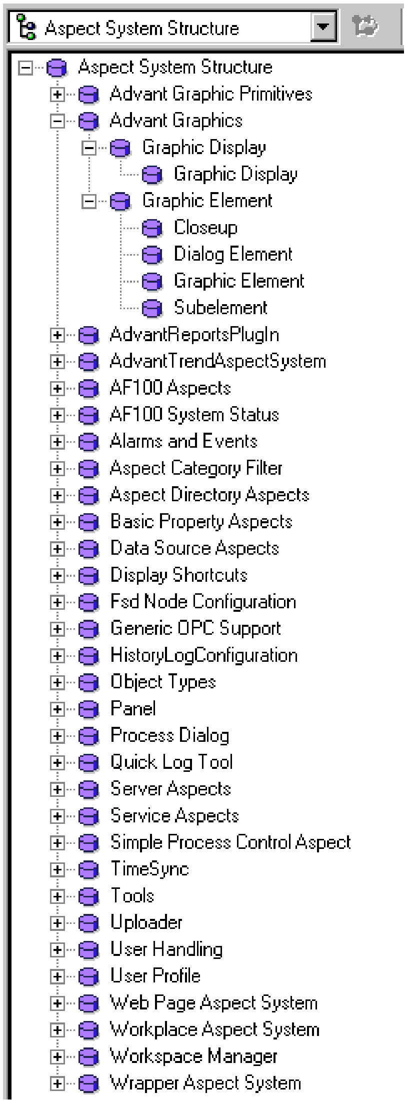
- Admininistrative Objects
  - Domains
    - Advant System
    - Default Domain
  - Inventory Object
    - Affinity
    - AfwAdminUser
    - AfwAdminUserGroup
    - AfwAdminUserPrivilege
      - Application Engineer Privilege
      - Operator Privilege
      - Software Developer Privilege
      - System Engineer Privilege
      - System Extensions Enable
    - Aspect Category
    - Aspect System
    - Aspect Type
    - Domain
    - ImplBind
    - Key
    - Node
    - Node Group
    - Operation
    - Permission
    - Priority
    - Service
    - Service Group
    - Service Provider

Figure B-1. Admin Structure

B.1.2 Aspect System Structure

The *Aspect System Structure* defines the aspect systems, aspect types and aspect categories that are available in a system. The structure shows the aspect systems, the corresponding aspect types, and under the aspect types their categories. The structure is primarily used to populate the New Aspect dialog used by the Plant

Explorer and other tools. See Figure B-2. The structure is typically updated by loading aspect system definitions (using add files).



- Aspect System Structure
  - Advant Graphic Primitives
  - Advant Graphics
    - Graphic Display
      - Graphic Display
    - Graphic Element
      - Closeup
      - Dialog Element
      - Graphic Element
      - Subelement
  - AdvantReportsPlugIn
  - AdvantTrendAspectSystem
  - AF100 Aspects
  - AF100 System Status
  - Alarms and Events
  - Aspect Category Filter
  - Aspect Directory Aspects
  - Basic Property Aspects
  - Data Source Aspects
  - Display Shortcuts
  - Fsd Node Configuration
  - Generic OPC Support
  - HistoryLogConfiguration
  - Object Types
  - Panel
  - Process Dialog
  - Quick Log Tool
  - Server Aspects
  - Service Aspects
  - Simple Process Control Aspect
  - TimeSync
  - Tools
  - Uploader
  - User Handling
  - User Profile
  - Web Page Aspect System
  - Workplace Aspect System
  - Workspace Manager
  - Wrapper Aspect System

Figure B-2. Aspect System Structure

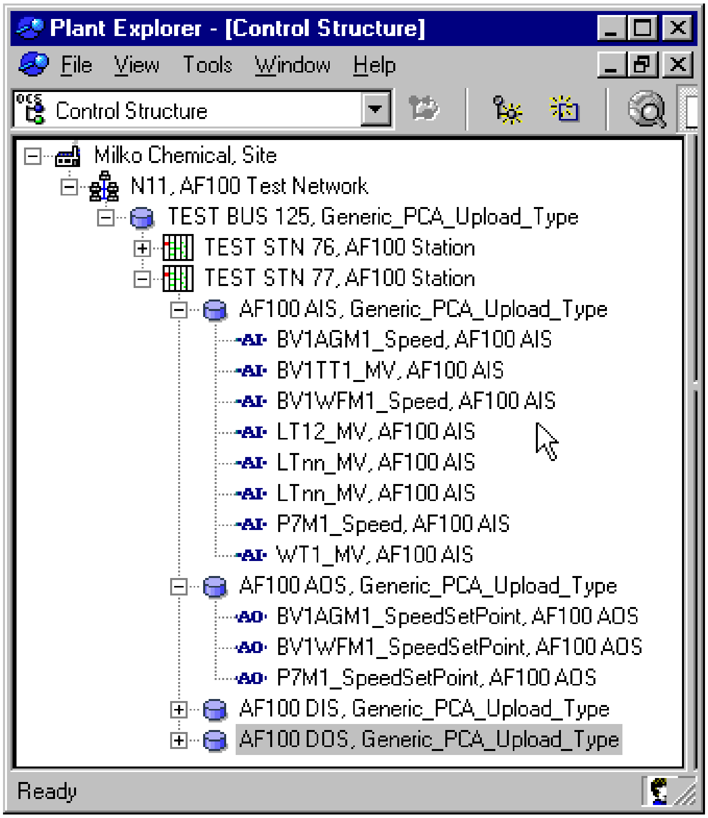
B.1.3 Control Structure

The *Control Structure* contains the networks and nodes that are used to control the process, that is the equipment of the Advant system.

The *Control Structure* is used to organize the process control environment in the system. This structure contains control networks, controllers, stations, different types of sub-objects (like I/O boards and process objects, and signals that are assigned to different objects within this structure).

The *Control Structure* is the base for the control operation of the system. In this structure the user can, for example, check consistency and system status.

The *Control Structure* is initialized without any objects. But the figure below gives an example of how the structure could look. See Figure B-3.



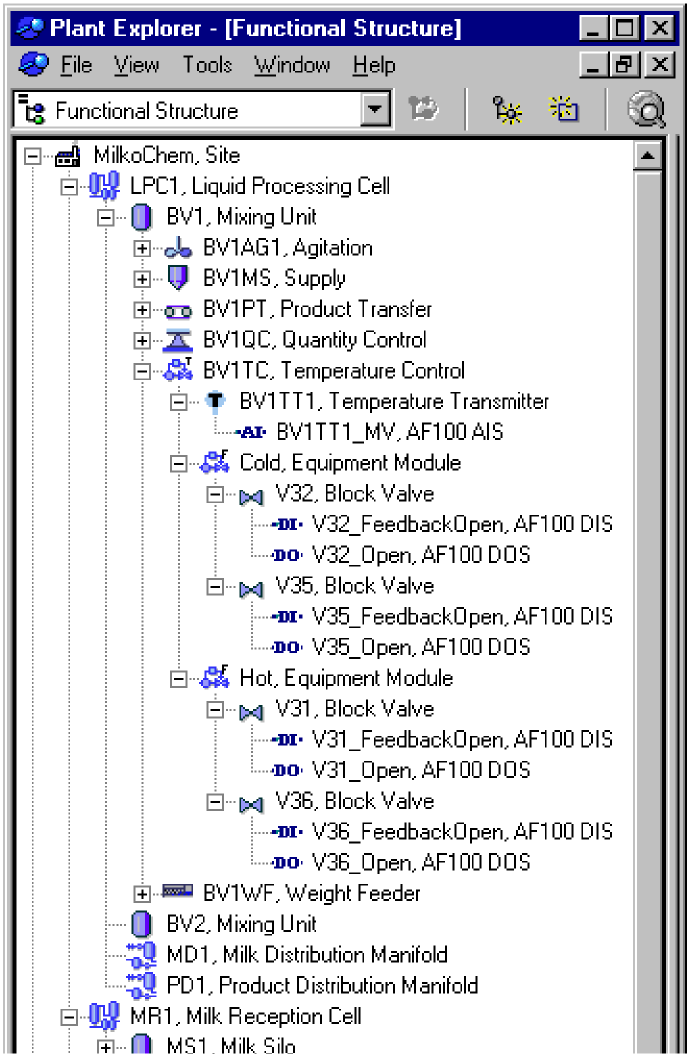
- Milko Chemical, Site
  - N11, AF100 Test Network
    - TEST BUS 125, Generic_PCA_Upload_Type
      - TEST STN 76, AF100 Station
      - TEST STN 77, AF100 Station
        - AF100 AIS, Generic_PCA_Upload_Type
          - BV1AGM1_Speed, AF100 AIS
          - BV1TT1_MV, AF100 AIS
          - BV1WFM1_Speed, AF100 AIS
          - LT12_MV, AF100 AIS
          - LTnn_MV, AF100 AIS
          - LTnn_MV, AF100 AIS
          - P7M1_Speed, AF100 AIS
          - WT1_MV, AF100 AIS
        - AF100 AOS, Generic_PCA_Upload_Type
          - BV1AGM1_SpeedSetPoint, AF100 AOS
          - BV1WFM1_SpeedSetPoint, AF100 AOS
          - P7M1_SpeedSetPoint, AF100 AOS
        - AF100 DIS, Generic_PCA_Upload_Type
        - AF100 DOS, Generic_PCA_Upload_Type

Figure B-3. Control Structure

B.1.4 Functional Structure

The *Functional Structure* is used to describe the functions of an installation site, for example, a plant. This structure is primarily built by the customer as he knows the structure of his site.

The *Functional Structure* is initialized without any objects. The figure below gives an example of how the Functional structure could look when objects have been added. See Figure B-4.



- MilkoChem, Site
  - LPC1, Liquid Processing Cell
    - BV1, Mixing Unit
      - BV1AG1, Agitation
      - BV1MS, Supply
      - BV1PT, Product Transfer
      - BV1QC, Quantity Control
      - BV1TC, Temperature Control
        - BV1TT1, Temperature Transmitter
          - BV1TT1_MV, AF100 AIS
        - Cold, Equipment Module
          - V32, Block Valve
            - V32_FeedbackOpen, AF100 DIS
            - V32_Open, AF100 DOS
          - V35, Block Valve
            - V35_FeedbackOpen, AF100 DIS
            - V35_Open, AF100 DOS
        - Hot, Equipment Module
          - V31, Block Valve
            - V31_FeedbackOpen, AF100 DIS
            - V31_Open, AF100 DOS
          - V36, Block Valve
            - V36_FeedbackOpen, AF100 DIS
            - V36_Open, AF100 DOS
      - BV1WF, Weight Feeder
    - BV2, Mixing Unit
    - MD1, Milk Distribution Manifold
    - PD1, Product Distribution Manifold
  - MR1, Milk Reception Cell
    - MS1, Milk Silo

Figure B-4. Functional structure

B.1.5 Graphics Structure

The *Graphics Structure* holds the graphic primitives and subelements available in the system. This structure is typically not accessed nor modified by a user as it is updated by different tools.

The structure uses the function of the Advant Graphics Builder. The structure is updated when subelements are created or modified, or when new primitives are loaded or unloaded.

The structure contains one root object, *Advant Graphics Tools.* The root object then holds three objects, one holding the subelements and one holding the primitives. The third object holds the other the graphical elements. See Figure B-5.

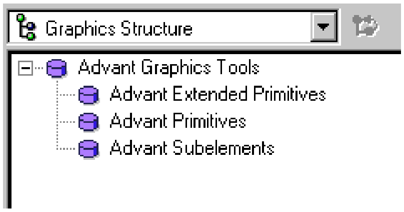
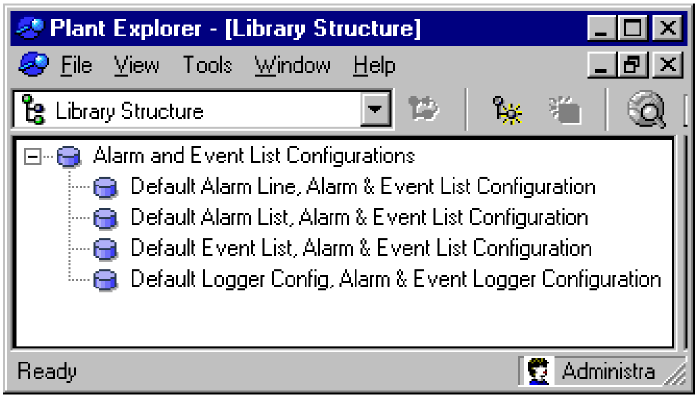


Figure B-5. Graphics Structure

B.1.6 Library Structure

In this release the usage of the *Library Structure* is limited to the storage of the default alarm and event aspect configurations.


Plant Explorer - [Library Structure]
File View Tools Window Help
Library Structure

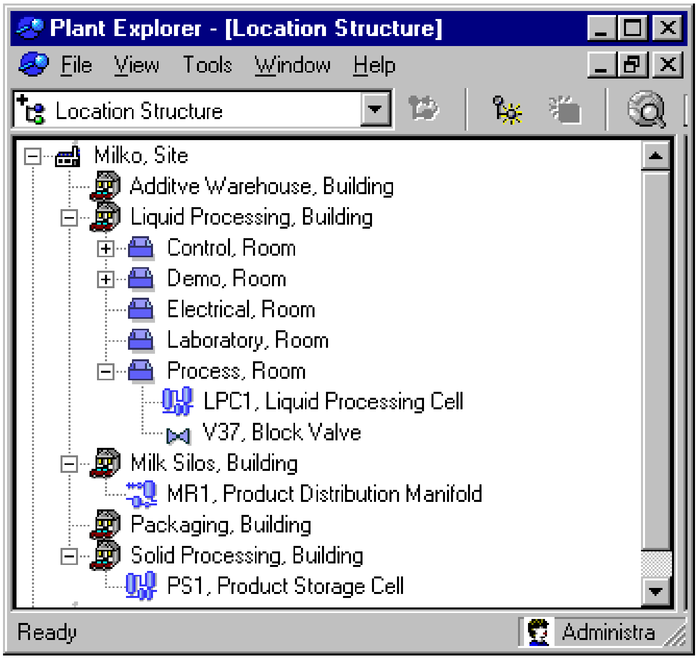
- Alarm and Event List Configurations
  - Default Alarm Line, Alarm & Event List Configuration
  - Default Alarm List, Alarm & Event List Configuration
  - Default Event List, Alarm & Event List Configuration
  - Default Logger Config, Alarm & Event Logger Configuration

Figure B-6. Library Structure

B.1.7 Location Structure

The *Location Structure* is supplied to help the user identify the geography or location of the objects in the plant. This structure must be built by the customer as only he has the knowledge of what buildings, rooms and locations that exist in the site.

At system start-up this structure contains no objects. The figure below gives an example of how the structure could look when containing objects. See Figure B-7.


Plant Explorer - [Location Structure]
File View Tools Window Help
Location Structure
Ready
Administra

- Milko, Site
  - Additve Warehouse, Building
  - Liquid Processing, Building
    - Control, Room
    - Demo, Room
    - Electrical, Room
    - Laboratory, Room
    - Process, Room
      - LPC1, Liquid Processing Cell
      - V37, Block Valve
  - Milk Silos, Building
    - MR1, Product Distribution Manifold
  - Packaging, Building
  - Solid Processing, Building
    - PS1, Product Storage Cell

Figure B-7. Location Structure

B.1.8 Node Administration Structure

The *Node Administration Structure* is similar to the *Control Structure*, but this structure is used for administration purposes. The main reason for this structure is that all nodes in the system must be found in at least one place and that nodes can be placed in multiple groups.

The structure contains two levels under the root: The first level contains node groups and the second level contains nodes.

The node group *All Nodes* has prebuilt support that disallows manual changes and updates automatically with all nodes in the system.

The nodes found in the *Node Administration Structure* are typically also found in the *Control Structure*. See Figure B-8.

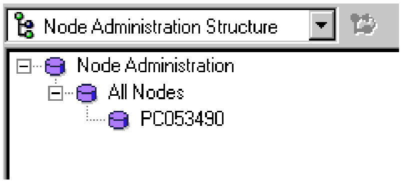
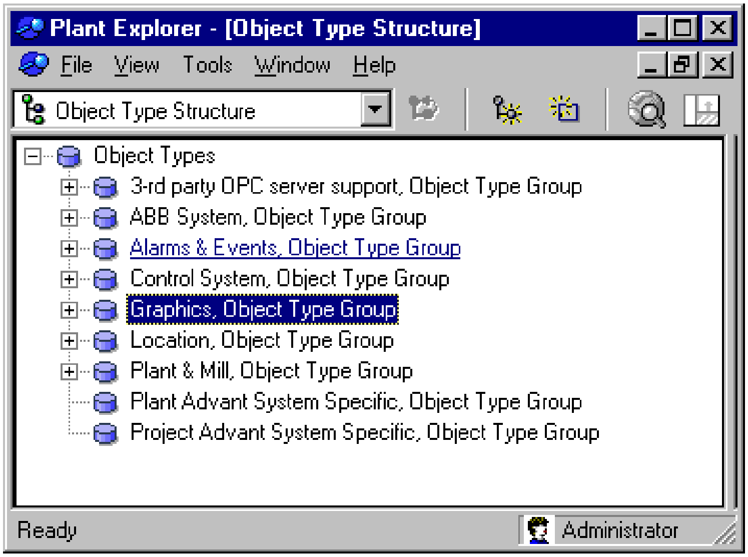


Figure B-8. Node Administration Structure

B.1.9 Object Type Structure

The *Object Type Structure* is used to organize the object types defined in the system. Object types allow easy creation and identification of objects. The user can, for example, create an *AI Process Object* and then get a set of aspects and substructures automatically.

The *Object Type Structure* contains Object Type objects or untyped objects. The untyped objects can be inserted to sort and group different object types. See Figure B-9.


Plant Explorer - [Object Type Structure]
File View Tools Window Help
Object Type Structure
Ready
Administrator

Object Types
- 3-rd party OPC server support, Object Type Group
- ABB System, Object Type Group
- Alarms & Events, Object Type Group
- Control System, Object Type Group
- Graphics, Object Type Group
- Location, Object Type Group
- Plant & Mill, Object Type Group
- Plant Advant System Specific, Object Type Group
- Project Advant System Specific, Object Type Group

Figure B-9. Object Type Structure

B.1.10 Service Structure

The *Service Structure* holds the services, their groups and providers, and the relations between these defined in a system. A service is a global function in the network, and the objects and aspects in the *Service Structure* control the definition and deployment of the services. See Figure B-10.

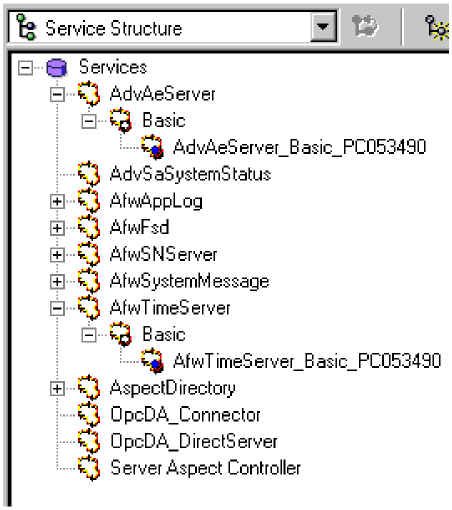
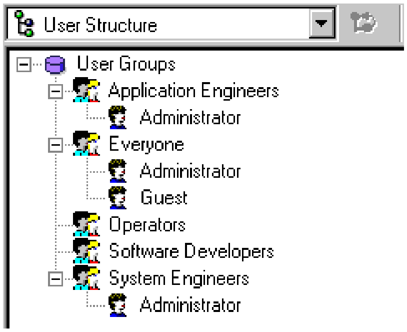


Figure B-10. Service Structure

B.1.11 User Structure

The *User Structure* holds the defined users and user groups that are allowed to work in the Advant system.

All Advant users must have a related Windows NT user account. The Advant user aspect primarily contains information about a user and what the user is allowed to do within the Advant system. It also contains data specific to the user, such as profile configuration values. The structure contains two levels under the root object: The first level holds groups, and the second holds users. The user group *Everyone* holds all users. Users can be added to groups by dragging and placing them into different user groups.



User Structure
- User Groups
  - Application Engineers
    - Administrator
  - Everyone
    - Administrator
    - Guest
  - Operators
  - Software Developers
  - System Engineers
    - Administrator

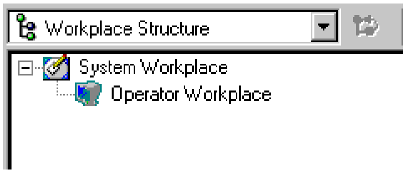
Figure B-11. User Structure

Only users and user groups objects can be placed in this structure. See Figure B-11.

B.1.12 Workplace Structure

The *Workplace Structure* holds the workplace objects defined in a system. The structure is used by the users to organize the workplaces that are in use. A workplace lower in the structure inherits definitions from higher level workplaces.

Only workplace objects are allowed in this structure. The types of aspect that can be placed here are numerous, and they control the layout and function of a workplace, for example tool bars and menu items. See Figure B-12.



Workplace Structure
- System Workplace
  - Operator Workplace

Figure B-12. Workplace Structure

INDEX

A

ABB Object 7
    Aspects of 14
    Build the Object Types 55
    Description 7
    Object types provided with the release 64
    Process Object Types 79
    Relations - structures 17
    Views of 15
ABB Object Type Concept 23
Advant Plant Explorer
    Description 7
    Window 10
Advant Users
    Description 7
    NT Users 163
Advant Workplace Window 10
Alarm and Event 139
    Alarm Band 149, 154
    Defaults 141
    Full Lists 145
Aspect
    Copy 78
    Description 8
    Edit 78

B

Bus Configuration Database
    BCD.MDB 123

C

Control Structure 21

F

Functional Structure 18

G

Graphic Displays
    Building 85
    Menu 159

H

History Log Aspects 167

L

Location Structure 19

M

Milko Chemical
    Example Project 27
    Overview 27

O

Object Type Structure 22

R

Relative name references
    In typical solution 40
    Relative name AdvantObject 40
    Relative Name Aspects 43
Restart OPC Server 124

S

Signal Object
    In control structure 125
Signal Objects 13
    In functional structure 131
    In location structure 135
Structures
    Admin Structure 206
    Aspect System Structure 207
    Control Structure 209
    Description 9

  - Functional Structure 210
  - General 205
  - Graphics Structure 212
  - Location Structure 213
  - Node Administration Structure 214
  - Object Type Structure 215
  - Service Structure 216
  - User Structure 217
  - Workplace Structure 218

T

Terminology 7
Trend aspects 170

U

Uploader 126

W

Web Aspects 172
Windows Application Aspects 175
Wrapper Aspects 174